RESOURCE MATERIALS COLLECTION
East-West Food Institute
East-West Center
1777 East-West Road
Honolulu, Hawaii 96822

FAO Fisheries Technical Paper No. 101 **FIRD/T101**
(Distribution restricted) **Stock Assessment**

Library
NOAA, National Marine Fisheries Service
SWFC Honolulu Laboratory F/SWC2
2570 Dole Street
Honolulu, HI 96822-2396

COMPUTER PROGRAMS FOR FISH STOCK ASSESSMENT

compiled by

Norman J. Abramson

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS
Rome, 1971

DOCUMENTS OF THE FISHERY RESOURCES DIVISION OF FAO DEPARTMENT OF FISHERIES

Documents which are not official FAO publications are issued in several series. They are given a restricted distribution and this fact should be indicated if they are cited. Most of them are prepared as working papers for meetings, or are summaries of information for use of member governments, organizations, and specialists concerned.

DOCUMENTS DE LA DIVISION DES RESSOURCES DES PÊCHES DU DÉPARTEMENT DES PÊCHES DE LA FAO

Des documents qui ne figurent pas parmi les publications officielles de la FAO sont publiés dans diverses séries. Ils font seulement l'objet d'une distribution restreinte, aussi convient-il de le préciser lorsque ces documents sont cités. Il s'agit le plus souvent de documents de travail préparés pour des réunions, ou de résumés d'information à l'intention des gouvernements des pays membres, ainsi que des organisations et spécialistes intéressés. Ces séries sont les suivantes:

DOCUMENTOS DE LA DIRECCION DE RECURSOS PESQUEROS DEL DEPARTAMENTO DE PESCA DE LA FAO

Esta Subdirección publica varias series de documentos que no pueden considerarse como publicaciones oficiales de la FAO. Todos ellos tienen distribución limitada, circunstancia que debe indicarse en el caso de ser citados. La mayoría de los títulos que figuran en dichas series son documentos de trabajo preparados para reuniones o resúmenes de información destinados a los Estados Miembros, organizaciones y especialistas interesados.

FAO Fisheries Report **FIR /R (No.)**
FAO Fisheries Circular **FIR /C (No.)**
FAO Fisheries Synopsis **FIR /S (No.)**

Special groups of synopses are identified by symbols followed by classification numbers based on indexed code of "Current Bibliography":

SAST Data concerning certain species and fish stocks.

MAST Information on methods and subjects.

OT Oceanographic data.

IT Limnological data.

and

CART Information concerning fisheries and resources of certain countries and regions (FID/S).

Des catégories spéciales de synopses sont identifiées à l'aide de symboles suivis des chiffres de classification basés sur le code d'indexation de la « Current Bibliography »:

SAST Données sur certaines espèces et populations de poissons.

MAST Renseignements sur des méthodes et des sujets.

OT Données océanographiques.

IT Données limnologiques.

et

CART Renseignements sur les pêcheries et les ressources de certains pays et régions (FID/S).

Grupos especiales de sinopsis se distinguen con las siglas siguientes, seguidas por números de clasificación que se basan en las claves de los índices de la « Current Bibliography »:

SAST Datos relativos a ciertas especies y poblaciones.

MAST Sinopsis sobre métodos y materias.

OT Sinopsis sobre oceanografía.

IT Sinopsis sobre limnología.

y

CART Información sobre los recursos acuáticos vivos de algunos países y regiones (FID/S).

FAO Fisheries Technical Paper **FIR /T (No.)**

Special groups of Technical Papers are identified by:

RE Indexed lists of experts and institutions drawn from Registers maintained by the Fishery Resources Division.

CB Lists of periodicals, special sections of "Current Bibliography for Aquatic Sciences and Fisheries," special bibliographies and papers concerning documentation problems.

MFS Provisional editions of "FAO Manuals in Fisheries Science."

Some documents also have another identification, if, for example, they have been contributed to a meeting for which papers have been numbered according to another system.

Des catégories spéciales de documents techniques sont identifiées à l'aide des symboles suivants:

RE Listes indexées d'experts et institutions tirées des registres tenus à jour par la Division des ressources des pêches.

CB Listes de périodiques, des sections spéciales de la « Current Bibliography for Aquatic Sciences and Fisheries », des bibliographies particulières et des articles sur les problèmes de documentation.

MFS Editions provisoires des « Manuels FAO de science halieutique ».

Certains documents portent parfois d'autres numéros d'identification, par exemple s'ils ont été préparés pour une réunion dont les documents ont été marqués à l'aide d'un autre système.

Grupos especiales de documentos técnicos se identifican por las siglas siguientes:

RE Listas índices de expertos y de instituciones tomadas de los registros que se llevan en la Dirección de Recursos Pesqueros.

CB Listas de periódicos, secciones especiales de la « Current Bibliography for Aquatic Sciences and Fisheries », bibliografías especiales y trabajos relativos a los problemas de documentación.

MFS Ediciones provisionales de los « Manuales de la FAO de Ciencias Pesqueras ».

Algunos documentos tienen también otra identificación si, por ejemplo, son contribuciones a una reunión cuyos documentos han sido marcados con arreglo a otros sistemas.

FAO FISHERIES TECHNICAL PAPER

This is one of a series of Technical Papers (see inside front cover) dealing with aspects of the work of the FAO Department of Fisheries, Fishery Resources Division, including the review of pertinent information on particular topics, to meet requests for assistance, for presentation to technical and scientific meetings, etc. Extra copies can be obtained by application to:

Research Information Section
Fishery Resources Division
FAO, Via delle Terme di Caracalla
Rome 00100, Italy

(Papers issued since January 1969)

FRs/T83	Manual of methods for fish stock assessment - Part V. The use of acoustic instruments in fish detection and fish abundance estimation	February 1969
FRi/T85	Directory of fish culture research institutions	February 1969
FRm/T71(Es)	Los pescadores y el tiempo	Marzo 1969
FRs/T84	Upwelling and fish production	June 1969
FRv/T87	Computer-aided studies of fishing boat hull resistance	December 1969
FRv/T88(Fr)	Le coefficient prismatique	Décembre 1969
(Es)	El coeficiente prismático	Diciembre 1969
FRv/T89 (Fr)	Types de bateau destinés aux pays dont les pêches sont en voie de développement	Décembre 1969
(Es)	Embarcaciones de pesca para las pesquerías en vías de desarrollo	Diciembre 1969
FRD/T90(Tri)	Careers in marine science - Carrières en sciences de la mer - Carreras en la ciencia marina	December 1969
FRv/T91	Some geotechnical and geophysical systems and their application to the planning of fishery harbour development programmes in developing countries	December 1969
FRs/T92	Fisheries management and the limitation of fishing	December 1969
(Fr)	L'aménagement des pêcheries et la limitation de la pêche	Août 1970
(Es)	La ordenación de las pesquerías y la limitación de la pesca	Marzo 1970
FRv/T93	Research fleet of the world	December 1969
FRi/T94	Facsimile of Section III - Fish, other aquatic life, and wildlife, of the Report of the Committee on Water Quality Criteria	December 1969
(Fr)	Section III. Poissons, autre vie aquatique, et faune sauvage. Traduction de l'original anglais du Rapport du « Committee on Water Quality Criteria »	Octobre 1970
(Es)	Facsímil de la Sección III - Peces, otras formas de vida acuática y fauna silvestre, del Informe del Comité sobre Criterios de Calidad del Agua	Diciembre 1969
FIRS/T70(Fr)	La notion de rendement maximal constant et l'aménagement des ressources halieutiques	Mars 1970
(Es)	El concepto del rendimiento máximo sostenible y la ordenación pesquera	Agosto 1970
FIRS/T97	The fish resources of the ocean	July 1970
FIRI/T86	Dissolved oxygen requirements of freshwater fishes	October 1970
FIRS/T98	International Biological Programme Section PM (Productivity Marine). Bibliography on methods of studying the marine benthos	October 1970
FIRI/T100	Reclamation of ponds, lakes, and streams with fish toxicants: A review	December 1970
FIRD/T101	Computer programs for fish stock assessment	February 1971

FAO Fisheries Technical Paper No. 101
(Distribution restricted)

FIRD/T101
Stock Assessment

COMPUTER PROGRAMS FOR FISH STOCK ASSESSMENT

compiled by

Norman J. Abramson
Consultant
Fishery Resources Division

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS
Rome, February 1971

PREPARATION OF THIS DOCUMENT

This handbook was prepared under the auspices of the FAO Fishery Data Centre and is intended primarily for use by FAO field projects and scientists collaborating with FAO. However, it is hoped that the computer program listings will be useful to any scientist who is engaged in fisheries stock assessment work and has access to a computer.

Readers are urged to contribute stock assessment programs which would have general application, for possible inclusion in future editions. Especially welcome would be versions or adaptations the programs presented here that could be executed on smaller-sized computers.

Mr. Norman Abramson compiled and edited the programs and operating instructions while on leave from the Operations Research Branch of the California Department of Fish and Game. He is presently on the staff of the U.S. National Marine Fisheries Service, at Terminal Island, California. The preparation and final typing of the manuscript was done by Miss Jessica Kwang, FAO.

DISCLAIMER

The programs contained in this publication are furnished without warranty. Neither the authors of the programs nor FAO Make any representation as to the accuracy or completeness of the information or programs and assume no responsibility with respect to their use.

Distribution:

FAO Department of Fisheries
FAO Regional Fisheries Officers
Selector SM
Selector SI
UNDP/SF and TA Experts
Author

Bibliographic entry:

Abramson, N.J. (Comp.) (1970)
FAO Fish.tech.Pap., (101):pag.var.
Computer programs for fish stock assessment

CONTENTS

		Page
PART 1:	USER'S GUIDE	
	INTRODUCTION	1
	PROGRAM USAGE	2
	Programming Language	2
	Contents of Program Operating Instructions	2
	Guidance for Non-programming Users	2
	Guidance for Programmers	3
	GENERAL PURPOSE STATISTICAL PACKAGES	3
	System/360 Scientific Subroutine Package	3
	BMD Biomedical Computer Programs	4
	CLASSIFICATION SYSTEM	4

PART 2: PROGRAM LISTINGS AND OPERATING INSTRUCTIONS

Name		Classification Number
AGECOM	- Age composition estimation	1.(-).1
BGC 2	- von Bertalanffy growth curve fitting	2.(-).1
BGC 3	- von Bertalanffy growth curve fitting	2.(-).2
BGC 4	- von Bertalanffy growth curve fitting	2.(5).3
FPOW	- Fishing power estimation	3.(-).1
MURPHY	- Fishing mortality estimation	4.(5).1
FRC 705	- Survival rate estimation	4.(1).2
FRC 701	- Yield curve with constant rates	7.(-).1
FRC 708	- Yield curve with varying rates	7.(-).2
YIELD 2	- Yield curve with varying rates	7.(-).3
GENPRD 2	- Generalized stock production model fitting	8.(-).1
POPSIM	- Population simulation and yield optimization	9.(10).1
RICKER 1	- Spawner-recruit curve fitting	10.(-).1
WTLN	- Weight-length curve fitting	11.(2).1
NORMSEP	- Normal distribution separator	11.(1).2

PART 1

USER'S GUIDE

INTRODUCTION

Over the past 15 years electronic computers have exerted a continual increasing and still accelerating influence on scientific research. The development and improvement of high level programming languages such as FORTRAM, ALGOL and PL/I have made these machines flexible and accessible to the non-specialist. In fisheries this has resulted in a number of programs designed specifically to solve quantitative problems associated with population dynamics and stock assessment.

We envisage a continued expansion of the application of computer methodology to fisheries problems with an attendant improvement in the efficiency of fisheries research. This improved efficiency will partly arise from the elimination of much time-consuming manual arithmetic work involved in problems such as calculating a weight-length curve. In this case, there will be no need for the fisheries researcher to look up thousands of logarithms in a table, enter them in a calculator, perform the computations, and then repeat the entire process to make sure there was no error. He will have more time to plan his program and analyze and integrate his results.

Problems which previously could not be solved at all because of the tremendous amount of computation required can be handled on computers. These include problems where no closed form of numerical solution is possible and only iterative operations yield the desired result. Iteration, of course, is a repetitive process where trial values are substituted into an equation and trial solutions and improved estimates are obtained until the equation is satisfied. When the equation is complex, a great quantity of computation is required to obtain a solution. Also included in this category of heavy computation problems are those which involve complex operations on large matrices. The number of separate calculations needed in such problems makes manual calculation not only too costly in time but also too inaccurate.

Another capability of computers that can make fisheries research more fruitful is their ability to rapidly summarize and tabulate voluminous quantities of numbers. It is not unusual to find fisheries organizations with large amounts of research data which have been only superficially analyzed because time and manpower have not been available to properly summarize the material. Once the initial programming has been accomplished, summarization and rapidly accessible storage of the data becomes routine and efficient.

A fourth and most recent area of computer application is simulation. Complex systems involving fish populations, fisheries, and related processing industries are programmed for repetitive trials on the computer. These systems, if they contain stochastic elements and feedback terms, are so complicated that analytical solutions are virtually unobtainable and computer modelling is the one way to observe outcomes over thousands of trials. Simulation models allow the fisheries manager to try various management strategies and examine their effects on the entire system.

This manual contains operating instructions and listings for a set of programs which could form the core of a program library useful for solving those problems commonly encountered when attempting to estimate population parameters or calculate yields obtainable from exploited fish stocks. While the scope of programs currently in use by fisheries workers is much broader than the set contained here, we have included only routines which are of a general nature; programs for processing log book data, research vessel observations and commercial landings information are usually applicable to alien situations only after substantial modification.

To expedite the preparation of this publication most of the material contained here was obtained from the program libraries of the Fisheries Research Institute, University of Washington and the California Department of Fish and Game. I wish to acknowledge Dr. Gerald J. Paulik and Mrs. Helen Bailey of the University of Washington and Miss Catherine L. Berude of the California Department of Fish and Game for their assistance in furnishing the listings and operating instructions which compose Part 2 of this manual. Also, acknowledgement is due to the programmers who originated the programs and operating instructions which appear here. Without their endeavours this manual would not have been possible.

It is planned that this collection of material will be expanded in the future and as additional programs are added they will be obtained from a wider range of sources.

PROGRAM USAGE

Programming Language

Almost all of the programs listed here are written in FORTRAN IV in the sense that they can be compiled without modification on the FORTRAN IV compilers of Control Data Corporation's 6000 series computers. They will require easily made modifications for use with FORTRAN IV compilers of other manufacturers. FORTRAN is the most widely available scientific programming language and, therefore, is the logical choice for programs which are to be broadly distributed. Future additions to this manual may include ALGOL and PL/I programs if those languages become generally adopted by computer manufacturers. However, assembly language programs will not be included because these languages differ substantially between computer manufacturers.

Contents of Program Operating Instructions

Operating instructions in each case contain a brief description of the mathematical and statistical procedures employed in the program or a citation to the relevant methodological paper. Also included in the instructions is a description of the input deck which gives the ordering and formats of the data cards, restrictions on the amount of input data and, when available, the approximate core storage requirements.

The operating instructions described above are generally the write-ups prepared by the program's author. In some cases we have appended notes describing ways of reducing core storage requirements so that the programs will be executable on smaller machines. Where non-standard FORTRAN statements are contained in the listings their function is discussed and substitutes are suggested. Those non-standard statements common to most of the programs are, however, described later in Part 1 rather than with the individual operating instructions.

Guidance for Non-programming Users

A person not conversant with computers who intends to employ the programs which follow will face some special problems: he does not understand how the computer system operates and must work in conjunction with a computer professional who uses strange jargon. Based upon personal experience, these factors may give the potential user a feeling of helplessness. To help alleviate this problem we will give a very brief description of the operation of a computer system and a procedure which usually ensures against the programmer making improper program alterations.

Before discussing the computer system, we wish to briefly mention commercial computer service bureaus. These organizations are located in many cities throughout the world and provide computer usage at hourly rates. They also provide professional programmers who will perform program modifications, original programming, and assist in the running of existent programs. The potential user may find it advantageous to employ some or all of the services these companies offer.

The names FORTRAN, ALGOL and PL/I have been mentioned previously and termed high-level languages. This refers to the fact that these languages are closer to the user's mathematical language than to instructions which the computer directly understands. A large computer program called a compiler automatically converts the high-level instructions into machine language and works out innumerable details relative to the storage of data and instructions within the machine's core memory. The number of machine language instructions generated from a FORTRAN program is much larger than the original number of FORTRAN statements.

Compilers are under control of another large program called the operating system. The operating system regulates the flow of work through the computer, controls a number of auxiliary programs such as the compilers, schedules peripheral devices such as printers, card readers and tape drives and performs bookkeeping chores. The programmer instructs the operating system with special punched cards called control cards. The user with programmer assistance need not be concerned with system control cards; no further mention of them appears in this manual.

A very important function for the user to perform when he uses a program listed in this book is the preparation of a comprehensive test problem and manually calculated results. These test data are to be used for the first run of the program to detect errors which may have occurred in modifying or key-punching the program or setting-up the data deck. Incidentally, this same procedure should be followed whenever one employs a programmer to prepare a new program. In addition to the test data procedure, the user should carefully examine the output from production runs for reasonable results. Frequently his subject matter knowledge of the expected range of results will detect faulty executions.

A note of caution to the user - be careful about employing computer programs if you do not have at least a moderate knowledge of the methodology which was programmed. To do otherwise is fraught with the danger of not understanding assumptions which underlie the methods.

Guidance for Programmers

Here we describe non-standard FORTRAN statements which occur frequently in the program listings. These statements will generally only be compatible with CDC 6000 series FORTRAN IV compilers.

The initial card of CDC programs is of the form: PROGRAM name (a, b, c, ...), where the arguments a, b, c, ... take forms such as: INPUT, OUTPUT, TAPE5 = INPUT, TAPE7 = OUTPUT. The preceding sequence implies that unit 5 is the system standard input unit, unit 7 is the standard output unit and unit 8 is employed by the program in other than a card reading or printing capacity. For compilation on other compilers this card will be removed or replaced according to the appropriate manufacturer's manual.

The statement IF (EOF,I) N1, N2 checks for an end-of-file on the previous read operation from unit I. If an end-of-file was encountered the program branches to statement N1; otherwise branching is to statement N2. Most FORTRAN IV compilers have an equivalent statement which can be substituted. Otherwise, a special card can be used to mark the end of the data deck and end-of-data branching carried out by an ordinary IF statement. Care should be taken that the location from which branching occurs is not changed when substitute statements are inserted.

The input and output statements READ N, list and PRINT N, list are found in many of the programs. The first statement reads from the system input unit, usually the card reader, according to format statement N and the second writes on the system output unit, ordinarily the line printer, according to format statement N. On some computers these will have to be replaced with the FORTRAN IV statements READ (I, N) list and WRITE (I, N) list.

Other exceptions to standard FORTRAN IV which will be found in some programs are the $, used to separate statements when more than one appears on a coding line and the multiple replacement expression which appears, for example, as X = Y = 3. * A. The multiple replacement statement above is equivalent to the expressions:

```
Y = 3. * A
X = Y
```

Finally, the alphabetic formats will be unsuitable for many computers because of word size. Many of the formats are multiples of A8 or A10. These may be easily converted to the word size appropriate to the object computer. For example, if using an IBM 360 with a word size allowing only A4, an 8A10 format in the listings may be replaced with a 20A4. Some of the programs contain 7-letter names for variables and subroutines instead of the FORTRAN IV standard of 6 letters.

GENERAL PURPOSE STATISTICAL PACKAGES

While standard statistical and mathematical procedures are frequently used for analysis of fisheries data, computer programs to carry out these analyses are not included here because packaged sets of such programs are widely available. We mention two of these program sets although others are available.

System/360 Scientific Subroutine Package

This is a set of more than 250 FORTRAN listings of subroutines for performing mathematical and statistical operations. The subroutines can be combined with a user's main program to perform a great variety of analytical procedures. Included are a number of sample main programs which illustrate the use of some of the subroutines and are essentially complete programs for certain types of statistical analyses.

The full title of this book is "System/360 Scientific Subroutine Package (360A-CM-03X) Version III, Programmers Manual". It is an IBM publication H20-0205-3 and is available from IBM branch offices throughout the world.

BMD Biomedical Computer Programs

The BMD series constitutes a comprehensive set of complete computer programs to solve many types of statistical problems.

A large amount of flexibility is allowed relative to data input formats and transformations of variables. Many of the programs contain printer plotting routines and thus provide graphical output.

A manual describing the programs and providing instructions for their use may be purchased. The actual programs are available from the Health Sciences Computing Facility, University of California, Los Angeles, California 90024, U.S.A. Almost all of the program contents are written in FORTRAN IV but sometimes modification will be necessary if they are used on other than large IBM 360 computers. Many computing service bureau organizations which handle scientific computing possess all or part of the BMD series.

The complete reference to the manual is: Dixon, W.J. 1968. BMD biomedical computer programs, University of California Press, Berkeley and Los Angeles, Cambridge University Press, London. X + 600 p.

CLASSIFICATION SYSTEM

Programs contained in Part 2 of this manual have been assigned primary and secondary classifications according to their area of application relative to stock assessment work. The classification scheme closely follows the section titles of: Gulland, J.A. 1969. Manual of methods for fish stock assessment. Part I. Fish population analysis. Rome, FAO. FAO Manuals in Fisheries Science No. 3. Users of the stock assessment manual should have little trouble relating these programs to the manual; however, in general the programs are not computing routines to perform the calculations described in the manual.

Numerical classification codes and the corresponding verbal descriptions are given below:

1. Sampling
2. Growth
3. Effort and catch/effort
4. Mortalities
5. Tagging
6. Recruitment and selection
7. Simple yield curve
8. Surplus production models
9. Other yield models
10. Density dependent effects
11. Miscellaneous

Each page of the program listings and accompanying operating instructions carry the following type of designation in an upper corner: A, (B), C, D. A is the primary classification, (B) is the secondary classification, C is the program number within the primary classification and D is the page number within a particular program. Page numbers for individual programs begin with the first page of the operating instructions and extend sequentially through the last page of the listing.

PART 2

PROGRAM LISTINGS AND OPERATING INSTRUCTIONS

Program name - AGECOM

Purpose

Estimates age composition using double sampling scheme with length as strata. Also provides estimates assuming simple random sampling of aged fish.

Programmer - D.J. Mackett

Originating organization - California Department of Fish and Game

Approximate storage - 19,500 (CDC 6600)

AGECOM, using length-age data collected in a double sampling scheme (Cochran 1953, Kutkuhn 1963, Mackett 1962), prints five tables containing the following statistics:

1. Relative age composition and 95% confidence limits of each percentage
2. Absolute age composition and 95% confidence limits of the number at each age
3. Age sample sizes that will reduce standard error of percentage estimate by 10 to 50 percent
4. Sample sizes for a minimum variance given a fixed sampling cost, and unit costs of ageing and measuring. The sample sizes are computed for both double sampling and simple random sampling; in addition, the expected standard errors using the computed sample sizes are printed for both sampling techniques
5. Relative length frequency table

A maximum of 20 age groups and 200 length strata can appear in the sample data.

More than one set of data can be analyzed during a single run by stacking the data decks. The results for a single set of data are usually computed in less than 10 seconds.

Under the double sampling scheme, the first sample is of lengths (length frequency) to estimate the length-strata sizes, the second or main sample is for ages. The second sample can be drawn (i) independently; (ii) as a subsample of the first; or (iii) as a subsample within length strata.

The computation of the variances from the sample is valid for any allocation (proportional, optimum, etc.) of the age sample within length strata. However, the estimated sample sizes and expected standard errors (Item 4) are computed for proportional sampling.

The data deck is arranged as follows:

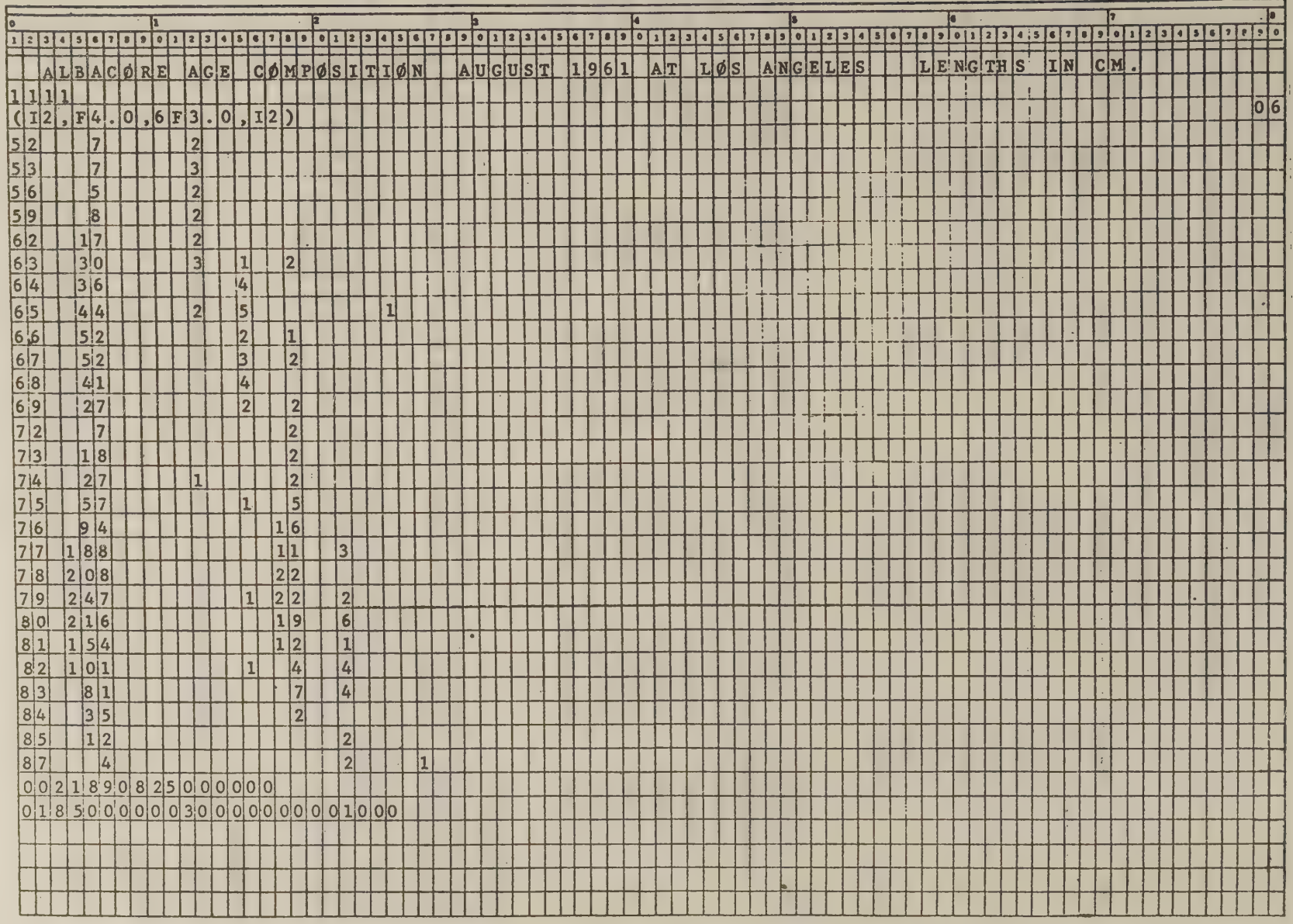
Card Number	Description and Format
1.	Columns 2-78 contain alpha-numeric characters which will appear as an identification or title on each page of the report.
2.	Control information, specifying which tables are to be compiled, is punched in columns 1-4. A 1 is punched to cause an operation to be performed; a zero causes the operation to be eliminated.

Card Column	Operation Control
1.	Number of fish at each age (absolute age composition).
2.	Estimated sample sizes to reduce standard error by 10 to 50 percent.
3.	Sample sizes for minimum variance with fixed cost.
4.	Relative length frequency table.

Card Number	Operation Control
3.	Columns 1–78 contain the variable format for the length–age data cards. The number of age groups appearing in the sample is punched in columns 79–80. The editing control symbols and their proper sequence are given in the following illustration: Column 1 (Iw, Fw.0, nFw.0, I2) .... Column 79–80 n The editing codes, read left to right, govern the edit of (i) length; (ii) length frequency; (iii) number in each age group; and (iv) "last card" indicator (see card 4 description). The w is an unsigned integer which specifies the width of a field, and n is the number of age groups in the sample.
4.	The fourth card in the deck is the first of those containing length–age data. A card appears for each length observed in the sample. However, a card should not appear for lengths which have no corresponding age data. The length–age data are punched in the following sequence: 1) length 2) length frequency 3) number in zero age group 4) number in one year age group . . . n+2) number in oldest age group n+3) last card indication (01 punched in last card; blank in all others) The number of age–group fields defined must equal the number obtained by adding 1 to oldest age in sample (see sample data).
5.	Columns 1–10 contain the total number of fish to be pro–rated by the percentages of the relative age composition. Columns 11–20 contain the variance of the total number of fish. (No decimal places appear in either field unless decimal point is punched.) Card 5 must be included in data deck if column 1 of card 2 is punched 1, otherwise, it must be eliminated.
6.	Columns 1–8 contain the cost of making a single age determination. Columns 9–16 contain the cost of making a single length measurement. Both numbers have six places to the right of the decimal point (which is not punched). Columns 17–24 contain the number of dollars available for sampling (fixed cost). Card 6 must appear if a 1 is punched in column 3 of card 2, otherwise, it must be eliminated.

Sample data and two of the resulting five output tables follow.

SAMPLE DATA FOR AGECØM



AGECOM

1.(-).1.3

AGECOM

ALBACORE AGE COMPOSITION, AUGUST 1961, AT LOS ANGELES
LENGTHS IN CM

ESTIMATED NUMBER OF FISH AT EACH AGE

AGE	NUMBER	VARIANCE	STD. ERROR	95% CONF. LIMITS LOWER	UPPER
0	0.	.000000000	0.	0.	0.
1	9743.	.357016309+07	1889.	6040.	13446.
2	27067.	.162948600+08	4037.	19155.	34979.
3	155018.	.518519331+08	7201.	140905.	169132.
4	26402.	.254597066+08	5046.	16512.	36291.
5	678.	.470291959+06	686.	666.	2022.
TOTAL	218908.	25000000.			

AGE	RELATIVE AGE COMPOSITION	VARIANCE COMPONENTS WITHIN STRATA	BETWEEN STRATA	VARIANCE	STANDARD ERROR	95% CONF. LIMITS LOWER	UPPER
0	.000000	.0000000	.0000000	.0000000	.000000	.000000	.000000
1	.044507	.5645138-04	.1705501-04	.7350639-04	.008574	.027703	.061311
2	.123644	.2942459-03	.3798969-04	.3322356-03	.018227	.087919	.159370
3	.708144	.7688835-03	.5196728-04	.8208507-03	.028650	.651989	.764299
4	.120606	.5098778-03	.1409528-04	.5239730-03	.022890	.075741	.165471
5	.003099	.9601270-05	.2128015-06	.9814071-05	.003133	.003042	.009239

Special notes

This program ends with an end-of-file check on input unit 5. On most computers the statement IF(EOF,5)5,4 can be removed and the program will operate properly with automatic termination when the end-of-file is reached.

Storage reduction can be accomplished if fewer than 200 length strata and/or fewer than 20 age groups are involved. Variables dimensioned 200 in COMMON and DIMENSION statements may be re-dimensioned to the maximum number of length strata needed. The same can be done for variables dimensioned 20 when fewer age groups will occur. If the length strata dimension is reduced, be sure to reduce the upper limit of the DO occurring in subroutine AGECMP at statement 22+2.

References

Cochran, W.G., 1963 Sampling techniques, 2nd ed. John Wiley and Sons, New York. 413 p.

Kutkuhn, J.H., 1963 Estimating absolute age composition of California salmon landings. Calif. Dept. Fish and Game, Fish. Bull. No. 120, 47 p.

Mackett, D.J., 1963 A method of sampling the Pacific albacore (Thunnus germo) catch for relative age composition. Proc.World Sci.Meet.Biol.Tunas & Rel.Sp.,FAO. Fish. Rpt. No. 6, Vol. 3.

AGECOM

```
      PROGRAM AGECOM (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON /TITLE1/TITLE(13)
      DIMENSION OPT(4)
      INTEGER OPT
      CALL HEAD(DUM)
    1 READ(5,2)(TITLE(I),I=1,13)
    2 FORMAT(13A6)
      IF(EOF,5) 5,4
    4 CALL TITL
      READ (5,3)(OPT(I),I=1,4)
      CALL AGECMP
      CALL RDMOUT
      IF(OPT(1).EQ.1) CALL NMBR
      IF(OPT(2).EQ.1) CALL REDUCE
      IF(OPT(3).EQ.1) CALL COST
      IF(OPT(4).EQ.1) CALL DATOUT
      GO TO 1
    3 FORMAT(7I1)
    5 STOP
      END


      SUBROUTINE HEAD(DUM)
      WRITE (6,30)
   30 FORMAT (1H1,16(/),29X, 67HDOUBLE SAMPLING FOR AGE COMPOSITION USIN
     1G LENGTH FOR STRATIFICATION,14(/),57X,16HDAVID J. MACKETT/ 55X,19H
     2STATE OF CALIFORNIA/51X,27HDEPARTMENT OF FISH AND GAME/51X,27HMARI
     3NE RESOURCES OPERATIONS/54X,21HBIOSTATISTICS PROGRAM/57X,15HTERMIN
     4AL ISLAND)
      RETURN
      END


      SUBROUTINE TITL
      COMMON/TITLE1/TITLE(13)
      WRITE(6,1) (TITLE(I),I=1,13)
    1 FORMAT(1H1,26X,13A6////)
      RETURN
      END


       SUBROUTINE AGECMP
      COMMON/ABLK/AGENO(20),P(20)/BBLK/COMP(20)/CBLK/VARP(20)
      COMMON/DBLK/F(200),L(200),RF(200)/EBLK/X(200,20)/FBLK/TOTF,VA(20)
      COMMON/NOST/M,N/GBLK/BTWN(20)
       DIMENSION FMT(13),LCONF(20),PAL(200,20),GAL(200,20),SUMX(200)
      DIMENSION SEP(20),UCONF(20),WITH(20)
      REAL  LCONF
      IFLG = 0
      READ(5,3)(FMT(I),I=1,13),N
    3 FORMAT(13A6,I2)
      WRITE (6,22)
   22 FORMAT (1H0,4X,3HAGE,8X,8HRELATIVE,14X,19HVARIANCE COMPONENTS,18X,
     A8HVARIANCE,10X,8HSTANDARD,6X,21H.95 CONFIDENCE LIMITS/1H ,17X,3HAG
     BE,13X,6HWITHIN,15X,7HBETWEEN,32X,5HERROR,10X,5HLOWER,7X,5HUPPER/
     C1H ,14X,11HCOMPOSITION,8X,6HSTRATA,16X,6HSTRATA)
      TOTF= 0.
      DO  5  I=1,200
      SUMX(I)=0.
      READ(5,FMT) L(I),F(I),(X(I,J),J=1,N),LC
      TOTF =TOTF + F(I)
      DO 13  J=1,N
```

```
   13 SUMX(I) = SUMX(I)+X(I,J)
      IF(LC-1) 5,10,5
    5 CONTINUE
   10 M=I
C       M IS NUMBER OF LENGTH STRATA
C  N IS NUMBER OF AGES IN SAMPLE
      DO 50 J=1,N
      AGENO(J)=0.0
      P(J)=0.0
      JA=J-1
      DO 51 I=1,M
   51 P(J)= P(J) +(F(I)/TOTF)*(X(I,J)/SUMX(I))
      IF(P(J)) 5100,5100,5101
 5100 P(J)=+0.0
      SEP(J)=+0.0
      WITH(J)=+0.0
      COMP(J)=+0.0
      VARP(J)=+0.0
      LCONF(J)=+0.0
      UCONF(J)=+0.0
      GO TO 50
 5101 VA(J)=0.0
      WITH(J)=0.
      BTWN(J)=0.
      DO 42 I=1,M
C RELATIVE LENGTH FREQUENCY
      RF(I) = F(I)/TOTF
C PROPORTION OF AGE J OF LENGTH I
      PAL(I,J)= X(I,J)/SUMX(I)
      QAL(I,J)= 1.-PAL(I,J)
C FOR COST EQUATION AND SAMPLE SIZE
      VA(J)=VA(J)+(RF(I)*PAL(I,J)*QAL(I,J))
C WITHIN STRATA VARIANCE COMPONENT.
      IF(SUMX(I)-1.) 38,35,38
   35 IFLG = 1
      GO TO 40
   38 WITH(J)=WITH(J) + ( (RF(I)*RF(I)*PAL(I,J)*QAL(I,J))/(SUMX(I)-1.) )
C  BETWEEN STRATA VARIANCE COMPONENT
   40 BTWN(J) =  BTWN(J) +(RF(I)*( (PAL(I,J)-P(J))**2 ))
   42 CONTINUE
C BETWEEN STRATA COMPONENT
      COMP(J) = BTWN(J)/TOTF
C VARIANCE OF AGE J PROPORTION
      VARP(J) = WITH(J) + COMP(J)
C STANDARD ERROR
      SEP(J) = SQRT(VARP(J))
C LOWER CONFIDENCE LIMIT
      LCONF(J) = P(J)-(1.96*SEP(J))
C UPPER CONFIDENCE LIMIT (LARGE N)
      UCONF(J) = P(J)+(1.96*SEP(J))
  500 WRITE(6,25) JA,P(J),WITH(J),COMP(J),VARP(J),SEP(J),LCONF(J),UCONF(
     1J)
  250 FORMAT(1H0,5X,I2,8X,F7.6,7X,E14.7,7X,E14.7,7X,E14.7,8X,F7.6,8X,
     1F8.6,5X,F8.6)
      IF(IFLG) 55,65
   55 WRITE (6,100)
  100 FORMAT (//// 99H *****  WARNING  *****   VARIANCE ESTIMATION INVAL
     1ID BECAUSE ONE OR MORE SUBSAMPLES WERE OF SIZE 1.)
   65 RETURN
      END
```

AGECOM

```
      SUBROUTINE  RANDOM
       COMMON/RBLK/VRAN(20),SERAN(20),TA(20),GTA
      COMMON/HBLK/PRAN(20)/EBLK/X(200,20)/NOST/M,N
      GTA=0.
      DO  3  J=1,N
      PRAN(J)= 0.
      VRAN(J)= 0.
      TA(J)=0.
      DO  2  I=1,M
    2 TA(J)= TA(J)+ X(I,J)
C TOTAL NO. AGED
    3 GTA = GTA + TA(J)
      DO 4 J=1,N
      PRAN(J)= TA(J)/GTA
      VRAN(J)=(PRAN(J)* (1.- PRAN(J)))/(GTA-1.)
    4 SERAN(J) = SQRT(VRAN(J))
      RETURN
      END


      SUBROUTINE DATOUT
      COMMON/DBLK/F(200),L(200),RF(200)/NOST/M,N/FBLK/TOTF,VA(20)
      CALL TITL
      MH=M/2
      WRITE (6,2) ( L(I),F(I),RF(I),L(I+MH),F(I+MH),RF(I+MH),I=1,MH)
    2 FORMAT (1H0,52X,21HLENGTH FREQUENCY DATA////1H0,26X,6HLENGTH,6X,
     A8HOBSERVED,6X,8HRELATIVE,10X,6HLENGTH,6X,8HOBSERVED,6X,8HRELATIVE/
     B39X,9HFREQUENCY,5X,9HFREQUENCY,21X,9HFREQUENCY,5X,9HFREQUENCY/
     C(1H ,28X,I4,F13.0,F13.4,13X,I4,F13.0,F13.4 ))
      IF(MOD(M,2)-1)10,6,10
    6 WRITE(6,11)L(M),F(M),RF(M)
   11 FORMAT(1H ,71X,I4,F13.0,F13.4)
   10 WRITE(6,7) TOTF
    7 FORMAT(1H0,71X,5HTOTAL, F14.0)
      RETURN
       END


      SUBROUTINE NM8R
      COMMON/ABLK/AGENO(20),P(20)/CBLK/VARP(20)/NOST/M,N
      DIMENSION LOWNO(20),SENO(20),UPNO(20),VARNO(20)
      REAL  LOWNO
      READ(5,1) SHNO,VSHNO
    1 FORMAT(2F10.0)
      DNO=SHNO
      DVNO=VSHNO
      CALL TITL
      DO 2J=1,N
      AGENO(J)= DNO* P(J)
      VARNO(J)=     (((DNO**2)*VARP(J))+((P(J)**2)*DVNO) -(VARP(J)*DVNO))
      SENO(J) = SQRT(VARNO(J))
      UPNO(J)= AGENO(J) + (1.96* SENO(J))
    2 LOWNO(J)= AGENO(J)- (1.96* SENO(J))
      WRITE(6,3)
   30FORMAT(1H0,46X,36HESTIMATED NUMBER OF FISH AT EACH AGE/// 31X,
     13HAGE,4X,6HNUMBER,7X,8HVARIANCE,6X,10HSTD. ERROR,6X,16H.95 CONF. L
     2IMITS/81X, 5HLOWER,7X,5HUPPER//)
      DO 4 J=1,N
      JA=J-1
    4 WRITE(6,5) JA,AGENO(J),VARNO(J),SENO(J),LOWNO(J),UPNO(J)
    5 FORMAT (1H ,32X,I2,F11.0,E18.7,3F13.0)
      WRITE(6,6) SHNO,VSHNO
    6 FORMAT(1H0,29X,5HTOTAL,2F12.0)
      RETURN
      END
```

```
      SUBROUTINE COST
      COMMON /HBLK/PRAN(20)/FBLK/TOTF,VA(20)/NOST/M,N/GBLK/BTWN(20)
      DIMENSION DEN(20),EXV(20),EXVR(20),SSA(20),SSN(20),SEX(20)
      DIMENSION SEXR(20)
      READ(5,50)CAF,CMF,CSF
      LCS=CSF
      CA=CAF
      CM=CMF
      CS=CSF
   50 FORMAT(F8.6,F8.6,F8.0)
      CALL TITL
      WRITE (6,51) LCS,CA,CM
   51 FORMAT (1H0,25X,8HDOUBLE SAMPLING WITH STRATIFICATION BY LENGTH C
     1OMPARED TO SIMPLE RANDOM SAMPLING///1H ,49X,31HSAMPLE SIZES NEEDED
     2 FOR MINIMUM/54X,24HVARIANCE WITH FIXED COST//65X,3HAND//50X, 31HE
     3XPECTED VARIANCES AND STANDARD/ 50X,31HERRORS USING THESE SAMPLE S
     4IZES//1H0,39X,1H$,I6,44H  AMOUNT AVAILABLE FOR SAMPLING (FIXED COS
     5T)//40X,1H$,F9.6,36H  COST OF A SINGLE AGE DETERMINATION//40X,1H$,
     6    F9.6,37H  COST OF A SINGLE LENGTH MEASUREMENT /// 1H0,10X,3HAG
     7E,18X,12HSAMPLE SIZES,18X,18HEXPECTED VARIANCES,14X,24HEXPECTED ST
     8ANDARD ERRORS//27X,4HD.S.,13X,6HS.R.S.,9X,4HD.S.,16X,6HS.R.S.,12X,
     94HD.S.,11X,6HS.R.S.//20X,7HLENGTHS,5X,4HAGES,8X,4HAGES //)
      SSR=CS/CA
      CALL RANDOM
      DO 5 J=1,N
      JA = J-1
      VA(J)=ABS(VA(J))
      BTWN(J)=ABS(BTWN(J))
      DEN(J) = (SQRT(VA(J))*SQRT(CA)) + (SQRT(BTWN(J))*SQRT(CM))
C SAMPLE SIZE AGES
      SSA(J) = ( (SQRT(VA(J))/SQRT(CA)) * (CS/DEN(J)) )
C SAMPLE SIZE LENGTH
      SSN(J) = ( (SQRT(BTWN(J))/SQRT(CM)) * (CS/DEN(J)) )
C EXPECTED DS VARIANCE
      EXV(J) = (DEN(J)**2)/CS
      EXVR(J)=   (PRAN(J)*(1.-PRAN(J))*CA)/CS
      SEX(J) = SQRT(EXV(J))
C EXPECTED RANDOM S.E.
      SEXR(J) = SQRT(EXVR(J))
    5 WRITE(6,10) JA,SSN(J),SSA(J),SSR,EXV(J),EXVR(J),SEX(J),SEXR(J)
   10 FORMAT(1H0,10X,I2,F13.0,F10.0,F12.0,E19.9,E20.9,F15.8,F15.8)
      RETURN
      END


      SUBROUTINE RDMOUT
      COMMON/HBLK/PRAN(20)/NOST/M,N/RBLK/VRAN(20),SERAN(20),TA(20),GTA
      CALL RANDOM
      CALL TITL
      WRITE (6,3)
    3 FORMAT (1H0,43X,32HSIMPLE RANDOM SAMPLING ESTIMATES//1H0,19X,41HAG
     1E   NUMBER   PROPORTION     VARIANCE OF,7X,28HSTANDARD      COEFFI
     2CIENT OF /50X,10HPROPORTION,9X,5HERROR,10X,9HVARIATION/84X,9H(PERC
     3ENT) //)
      DO 5 J=1,N
      IF(PRAN(J)) 30,30,31
   30 RELSE = +0.0
      GO TO 32
   31 RELSE = SERAN(J)/PRAN(J)
   32 JA=IABS(J-1)
    5 WRITE(6,6) JA,TA(J),PRAN(J),VRAN(J),SERAN(J),RELSE
    6 FORMAT(1H0,19X,I2,F10.0,3X,F9.7,4X,E14.7,3X,E14.7,6X,F9.4)
      WRITE(6,7) GTA
    7 FORMAT(1H0,19X, 5HTOTAL,F7.0)
      RETURN
      END
```

```
      SUBROUTINE REDUCE
      COMMON/BBLK/COMP(20)/CBLK/VARP(20)/FBLK/TOTF,VA(20)/NOST/M,N
      DIMENSION AR(20,5),LAR(20,5)
      CALL TITL
      WRITE (6,71) TOTF
   71 FORMAT (1H0,35X,59HESTIMATED AGE SAMPLE SIZES NEEDED TO REDUCE STA
     1NDARD ERRORS/45X,42H10 TO 50 PERCENT-ASSUMING LENGTH-FREQUENCY/39X
     2,54HSAMPLE SIZE REMAINS CONSTANT AND PROPORTIONAL SAMPLING  ///
     3 1H0,15X,18HLENGTH SAMPLE SIZE,F8.0,11H (CONSTANT) //// 1H0,50X,
     4 17HPERCENT REDUCTION///37X,3HAGE,7X,2H10,9X,2H20,9X,2H30,9X,2H40,
     5 9X,2H50///)
      DO 61  J=1,N
      KK=0
      JA=J-1
      DO  62  I=1,5
      D=I
      FF= (1.-.1*D)**2
      IF(FF*VARP(J)-COMP(J))62,62,50
   50 KK=KK+1
C SAMPLE SIZE TO REDUCE VARIANCE
      AR(J,I) = VA(J)/(FF*VARP(J)-COMP(J))
   62 LAR(J,I)=IFIX(AR(J,I) )
      IF(KK)64,64,10
   10 WRITE (6,63) JA,(LAR(J,I),I=1,KK)
      GO TO (65,66,67,68,61),KK
   64 WRITE(6,641) JA
  641 FORMAT(1H036X,I2,5X,5(3H  ^,8X))
      GO TO 61
   65 WRITE(6,651)
  651  FORMAT(1H+,54X,4(3H  ^,8X))
      GO TO 61
   66 WRITE(6,661)
  661 FORMAT(1H+,65X,3(3H  ^,8X))
      GO TO 61
   67 WRITE(6,671)
  671 FORMAT(1H+,76X,2(3H  ^,8X))
      GO TO 61
   68 WRITE(6,681)
  681 FORMAT(1H+,87X,3H  ^)
   61 CONTINUE
   63 FORMAT(1H0,36X,I2,5X,5(I6,5X))
      WRITE(6,81)
   81 FORMAT(1H0 ,3X,///5X,91H ^ ... STANDARD ERROR CANNOT BE REDUCED BY
     1 THIS PERCENTAGE WITHOUT INCREASING LENGTH SAMPLE         )
      RETURN
      END
```

Program name - BGC 2

Purpose

Fits the von Bertalanffy growth-in-length curve to equally spaced age groups with unequal sample sizes for separate ages.

Programmer - Norman J. Abramson

Originating organization - California Department of Fish and Game

Approximate storage - 50,700 (CDC 6600)

This routine fits the curve $l_t = L_\infty \left[1 - \exp\left(-kt+kt_o\right)\right]$ using the least squares procedure described by Tomlinson and Abramson (Calif. Dept. of Fish and Game, Fish Bull. 116, p. 11, 1961) with weights proportional to the sample sizes. The program always yields estimates when a least squares solution exists and immediately terminates the run when there is no solution. In this respect, it is superior to a previously described program (Trans. Amer. Fish. Soc. 92(3):310, 1963) which occasionally does not converge to estimates even when a solution exists. A constant time interval between ages is required but the numbers of lengths in the age groups may be unequal. The output provides estimates of L_∞, k, t_o, and their standard errors, fitted lengths, sample mean lengths and their standard errors, the variance-covariance matrix and the standard error of estimate.

A single curve can be fitted in about 10 seconds and the estimates are accurate to at least four significant digits. Any number of curves can be computed on one run by stacking the data decks. A maximum of 30 age groups with at least 2 and at most 1500 lengths in each group can be handled.

Four or more cards of control information precede the cards containing length data.

Card 1	Columns 2-78 contain alphanumeric characters which constitute a sub-title on the report.
Card 2	Variable format for the length data.
Card 3	Number of age groups (cols. 1 and 2), age of the youngest age group (cols. 3-6), time interval between age groups (cols. 7-10). The latter two numbers each have three digits to the right of the decimal point (unpunched)
Card 4, 4A	Number of lengths in each age group (4 cols. per age group) arranged in order of ascending age. If more than 20 age groups are used, the numbers continue onto a second card.
Card 5, and remainder of deck	Length data punched according to the format on card 2. Age groups are arranged in ascending order and each age group begins on a new card.

A sample set of data and the resultant output are attached.

VON BERTALANFFY GROWTH IN LENGTH CURVE

BGC 2 SAMPLE PROBLEM TEST DECK

ESTIMATED PARAMETERS AND STANDARD ERRORS

	L INFINITY	K	T SUB-ZERO
ESTIMATES	1102.17	.190181	-2.3232
STANDARD ERRORS	90.84	.049441	.574846

BGC 2

FITTED LENGTHS AND SAMPLE LENGTHS

AGE	FITTED LENGTH	SAMPLE MEAN LENGTH	S.E. OF SAMPLE MEAN	SAMPLE SIZE
.0	393.62	NO SAMPLE DATA FOR THIS AGE		
1.0	516.34	512.00	17.302	9
2.0	617.80	631.27	15.737	11
3.0	701.69	671.83	18.582	6
4.0	771.04	770.56	9.982	9
5.0	828.39	848.75	11.747	8
6.0	875.81	869.43	13.571	7
7.0	915.01	893.80	17.028	5
8.0	947.43	978.50	14.500	2

SAMPLE DATA BEYOND AGE 8.0 NOT AVAILABLE

VARIANCE-COVARIANCE MATRIX

	L INFINITY	K	T SUB-ZERO
L INFINITY	.82519931+04	-.44147856+01	-.47356272+02
K	-.44147856+01	.24443924-02	.27419964-01
T SUB-ZERO	-.47356272+02	.27419964-01	.33044840-00

STANDARD ERROR OF ESTIMATE 43.7393

PROGRAMMED BY BIOMETRICAL ANALYSIS UNIT. MRO. CALIF. FISH AND GAME, MAY, 1964- -BGC 2

BGC 2 SAMPLE PROBLEM TEST DECK

```
(11F4.0)
0810001000
000900110006000900080007000500002
048005480503048505950497058304420475
0636068806990652066205540559066806450559062 2
0659064506660662063707 62
0735083707770748077207907490764 0763
0887084308580894084107870846 0834
0834084409340882089208390861
0855091509400905085 4
09930964
```

Special notes

This program checks for an end-of-file on input unit 5 at statement 2 + 1. Satisfactory results will be obtained on most computer systems if this statement is deleted.

A reduction in storage of about 21,000 words can be attained by following the instructions for dimension changes in the comment cards (C in column 1) at the beginning of the main program. This change reduces the program's capacity to 600 lengths for each of 30 age groups. Further reductions in this dimension statement will save additional storage at the cost of data handling capacity.

Subroutine NVERT checks for an overflow condition by calling OVERFL, a built-in FORTRAN IV library subroutine. If OVERFL is not available and overflow is of no concern, insert the statement KI = 2 at the beginning of NVERT and remove all CALL OVERFL (KI) statements.

BGC 2

On rare occasions this program, when run on a machine with a 36-bit or smaller word size, has failed to find a solution when, in fact, one exists. This is due to digital problems. If graphical analysis indicates a solution exists but the program fails to find it, one can try successively increasing the value of E to 0.01 and 0.1 and lowering the value of B to 0.99 and 0.9. These changes would be made in statements 25+1 and 25+2 in BGC 2.

BGC 2

```
      PROGRAM BGC2 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     BGC PROGRAM II(METHOD OF FISH BULLETIN 116)
      DOUBLE PRECISION WL,WTZM,WZ2M,WZM,WTZ2M,WLTZM,SUMW,WLZM
      DIMENSION TITLE(13),FMAT(13),NA(30),FNA(30),T(30),SUM(30),
     1          AVLGTH(30),A(3,3),EXPECT(30),EMNKT(30),LL(10),
     2SEMEAN(30),WT(30),M(30),FLNGTH(30,1500)
C     THIS DIMENSION STATEMENT IS FOR A 64K MACHINE. FOR A 32K REPLACE
C     THE CARD ABOVE WITH--
C    2SEMEAN(30),WT(30),M(30),FLNGTH(30,600)
    3 FORMAT(13A6)
    2 READ (5,3)(TITLE(I),I=1,13)
      IF(EOF,5) 999,998
  998 READ (5,3) (FMAT(I),I=1,13)
4     FORMAT(20I4)
      READ (5,5)K,T1,Q
C     K IS NO. OF AGE GROUPS , T1 IS AGE OF YOUNGEST GROUP , Q IS TIME
C     INTERVAL BETWEEN GROUPS
5     FORMAT(I2,2F4.3)
      READ (5,4)(NA(I),I=1,K)
C     NA(I) IS NO. OF FISH IN ITH AGE GROUP
      TOTALN =0.0
      DO 10 I=1,K
      SUM(I)= 0.0
      N= NA(I)
      READ  (5,FMAT)(FLNGTH(I,J),J=1,N)
      DO 6 J=1,N
    6 SUM(I)= SUM(I)+FLNGTH (I,J)
      FNA(I)= NA(I)
      M(I)= I-1
      T(I)=FLOAT(M(I))*Q
      TOTALN= TOTALN+ FNA(I)
   10 AVLGTH(I)= SUM(I)/FNA(I)
      DO 25 I=1,K
   25 WT(I)= FNA(I)/TOTALN
      E=0.001
      B=0.999
      CALL VALUE (Y,WT,AVLGTH,T,M,E,K,WL,WTZM,WZ2M,WZM,WTZ2M,WLTZM,
     1SUMW,WLZM)
      CALL VALUE (X,WT,AVLGTH,T,M,B,K,WL,WTZM,WZ2M,WZM,WTZ2M,WLTZM,
     1SUMW,WLZM)
      IF(SIGN(1.,Y)-SIGN(1.,X))29,500,29
   29 IF (.0001-(B-E))34,46,46
   34 C= (B+E)/2.
      CALL VALUE(U,WT,AVLGTH,T,M,C,K,WL,WTZM,WZ2M,WZM,WTZ2M,WLTZM,
     1SUMW,WLZM)
      IF(SIGN(1.,U)-SIGN(1.,Y))49,59,49
   59 Y=U
      E=C
      GO TO 29
   49 B=C
      X=U
      GO TO 29
   46 Z=E-(B-E)*Y/(X-Y)
      CALL VALUE (X,WT,AVLGTH,T,M,Z,K,WL,WTZM,WZ2M,WZM,WTZ2M,WLTZM,
     1SUMW,WLZM)
      D= SUMW* WZ2M- WZM**2
      FLNF=(1./D)*(WL*WZ2M-WZM*WLZM)
      B1=(1./D)* (WL*WZM-SUMW*WLZM)
      FK=(1./Q)*(0.-ALOG(Z))
      FTO=(1./FK)*(ALOG(B1/FLNF))
      DO 11 I=1,3
      DO 11 J=1,3
```

BGC 2

```
 11 A(I,J)= 0.0
    DO 42 I=1,K
    EMNKT(I)=EXP(0.0-FK*(T(I)-FTO))
    A(1,1)= A(1,1) + FNA(I)* (1.0- EMNKT(I))**2
    A(1,2)= A(1,2)+ FLNF* FNA(I)*(EMNKT(I)-EMNKT(I)**2)*(T(I)- FTO)
    A(1,3)= A(1,3)- FK*FLNF *FNA(I)*(EMNKT(I)-EMNKT(I)**2)
    A(2,2)= A(2,2)+ FNA(I)*(FLNF*EMNKT(I)*(T(I)- FTO))**2
    A(2,3)=A(2,3)-FNA(I)*FK*(FLNF*EMNKT(I))**2*(T(I)-FTO)
 42 A(3,3)= A(3,3)+ FNA(I)* (FK*FLNF*EMNKT(I))**2
    A(2,1)= A(1,2)
    A(3,1)= A(1,3)
    A(3,2)= A(2,3)
    CALL NVERT(3,3,A,LL,DET,KER)
    IF(KER.GT.1) GO TO 505
    SESQ= 0.0
    DO 55 I= 1,K
    EXPECT(I)= FLNF* (1.0-EMNKT(I))
    N=NA(I)
112 DO 55 J=1,N
 55 SESQ=SESQ+(FLNGTH(I,J)-EXPECT(I))**2
    SESQ=SESQ/(TOTALN-3.0)
    DO 60 I=1,3
    DO 60 J=1,3
 60 A(I,J)=A(I,J)*SESQ
 61 FORMAT(1H1,16X,38HVON BERTALANFFY GROWTH IN LENGTH CURVE,//)
    WRITE (6,61)
    WRITE (6,3)(TITLE(I),I=1,13)
 62 FORMAT(1H0,15X,40HESTIMATED PARAMETERS AND STANDARD ERRORS)
    WRITE (6,62)
 63 FORMAT(1H0,16X,10HL INFINITY,9X,1HK,9X,10HT SUB-ZERO)
    WRITE (6,63)
 64 FORMAT(1H ,9HESTIMATES,7X,F10.2,6X,F10.6,5X,F7.4)
    FTO=FTO+T1
    WRITE (6,64)FLNF,FK,FTO
    SEL=SQRT(A(1,1))
    SEK=SQRT(A(2,2))
    SETO=SQRT(A(3,3))
 65 FORMAT(1H ,15HSTANDARD ERRORS,1X,F10.2,7X,F9.6,4X,F10.6)
    WRITE (6,65)SEL,SEK,SETO
 66 FORMAT(1H0,19X,33HFITTED LENGTHS AND SAMPLE LENGTHS)
    WRITE (6,66)
 67 FORMAT(1H0,4H AGE,2X,13HFITTED LENGTH,2X,18HSAMPLE MEAN LENGTH,2X,
   119HS.E. OF SAMPLE MEAN,2X,11HSAMPLE SIZE)
    WRITE (6,67)
115 DO 68 I=1,K
 68 SEMEAN(I)=0.0
116 DO 70 I=1,K
    N=NA(I)
    T(I)=T(1)+T1
117 DO 69 J=1,N
 69 SEMEAN(I)=SEMEAN(I)+(FLNGTH(I,J)-AVLGTH(I))**2
 70 SEMEAN(I)=SQRT(SEMEAN(I)/(FNA(I)*(FNA(I)-1.0)))
118 IF(AINT(T(1))-T(1))71,72,72
 71 LAGE= T(1)+1.0
    GO TO  73
 72 LAGE=T(1)
119 IF(LAGE-0)90,90,73
 73 DO 75 1=1,LAGE
    FI=I-1
    EMNKFI=EXP(0.0-FK*(FI-FTO))
    FL= FLNF*(1.0-EMNKFI)
 74 FORMAT(1H0,F4.1,2X,F10.2,5X,27HNO SAMPLE DATA FOR THIS AGE)
 75 WRITE (6,74)FI,FL
```

```
   76 FORMAT(1H ,F6.3,1X,F10.2,8X,F10.2,11X,F11.3,10X,I4)
   90 WRITE (6,76)(T(I),EXPECT(I),AVLGTH(I),SEMEAN(I),NA(I),I=1,K)
   77 FORMAT(1H ,23HSAMPLE DATA BEYOND AGE ,F4.1,14H NOT AVAILABLE)
      WRITE (6,77)T(K)
   78 FORMAT(1H0,23X,26HVARIANCE-COVARIANCE MATRIX)
      WRITE (6,78)
      WRITE(6,161)
161   FORMAT(1H0,20X,10HL INFINITY,10X,1HK,9X,10HT SUB-ZERO)
   79 FORMAT(1H ,10HL INFINITY,5X,E15.8,3X,E15.8,2X,E15.8)
      WRITE (6,79)(A(1,J),J=1,3)
   80 FORMAT(1H ,1HK,14X,E15.8,3X,E15.8,2X,E15.8)
      WRITE (6,80)(A(2,J),J=1,3)
   81 FORMAT(1H ,10HT SUB-ZERO,5X,E15.8,3X,E15.8,2X,E15.8)
      WRITE (6,81)(A(3,J),J=1,3)
      SEEST=SQRT(SESQ)
   91 FORMAT(1H0,26HSTANDARD ERROR OF ESTIMATE--,F11.4)
      WRITE (6,91)SEEST
      WRITE (6,503)
  503 FORMAT (1H0,10HPROGRAMMED,1X,2HBY,1X,11HBIOMETRICAL,1X,8HANALYSIS,
     11X,5HUNIT,1X,4HMRO,1X,6HCALIF.,1X,4HFISH,1X,3HAND,1X,5HGAME,1X,4HM
     2AY,1X,6H1964--,6HBGC II)
      GO TO 2
   82 FORMAT(1H1)
500   WRITE (6,82)
      WRITE (6,3)(TITLE(I),I=1,13)
      WRITE (6,502)
  502  FORMAT (1H0,60HTHERE IS NO SOLUTION FOR Z = EXP(-KQ) BETWEEN .001
     1AND .999.)
      GO TO 2
505   WRITE(6,82)
      WRITE (6,3)(TITLE(I),I=1,13)
      WRITE (6,506)
  506  FORMAT (1H0,48HDATA SET DELETED--MATRIX INVERSION NOT POSSIBLE.)
      GO TO 2
  999 STOP
      END


      SUBROUTINE VALUE (ROOTZ,WT,BRL,T,M,Z,N,WL,WTZM,WZ2M,WZM,WTZ2M,
     1WLTZM,SUMW,WLZM)
      DOUBLE PRECISION WL,WTZM,WZ2M,WZM,WTZ2M,WLTZM,SUMW,WLZM
      DIMENSION WT(30),BRL(30),T(30),M(30)
      WL=0.0
      WTZM=0.0
      WZ2M=0.0
      WZM=0.0
      WTZ2M=0.0
      WLTZM=0.0
      SUMW= 0.0
      WLZM=0.0
      DO 14 J=1,N
      WL=WL+WT(J)*BRL(J)
      WTZM=WTZM+WT(J)*T(J)*Z**M(J)
      WZ2M=WZ2M+WT(J)*Z**(2*M(J))
      WZM=WZM+WT(J)*Z**M(J)
      WTZ2M=WTZ2M+WT(J)*T(J)*Z**(2*M(J))
      WLTZM=WLTZM+WT(J)*BRL(J)*T(J)*Z**M(J)
      SUMW=SUMW+WT(J)
   14 WLZM=WLZM+WT(J)*BRL(J)*Z**M(J)
      ROOTZ = WL*(WTZM*WZ2M-WZM*WTZ2M)- WLTZM*(SUMW*WZ2M- WZM **2)+
     1WLZM*(SUMW*WTZ2M- WZM*WTZM)
      RETURN
      END
```

BGC 2

```
      SUBROUTINE NVERT (M,MX,W,LW,DET,KER)
C     INVERT A M BY M MATRIX, W, STORED IN A MX BY MX ARRAY
C     LW IS A 1-DIMENSIONAL ARRAY OF AT LEAST 2*M ELEMENTS
C     KER IS AN ERROR FLAG -- =1 NORMAL,=2 SINGULAR MATRIX,=3 OVERFLOW
      DIMENSION W(3,3),LW(10)
      MX = M
    1 DET=1.
      DO 10 L=1,M
      LW(L) = L
      ML=M+L
   10 LW(ML)=L
      DO 150  K= 1,M
      IBIG=K
      JBIG = K
      DO 15 I= K,M
      DO 15 J= K,M
      IF (ABS (W(IBIG,JBIG))-ABS (W(I,J)))  13,15,15
   13 IBIG = I
      JBIG = J
   15 CONTINUE
      IF (W(IBIG,JBIG))  19,16,19
   16 KER = 2
      RETURN
   19 IF (JBIG - K) 20,24,20
   20 DO 22 I = 1,M
      WTRAN = W(I,K)
      W(I,K) = W(I, JBIG)
   22 W(I,JBIG) = WTRAN
      LTRAN=LW(JBIG)
      LW(JBIG) = LW(K)
      LW(K) = LTRAN
   24 IF (IBIG - K) 25,100,25
   25 DO 30 J=1,M
      WTRAN = W(IBIG,J)
      W(IBIG,J) = W(K,J)
   30 W(K,J) = WTRAN
      MIBIG=M+IBIG
      LTRAN=LW(MIBIG)
      MK=M+K
      LW(MIBIG)=LW(MK)
      LW(MK)=LTRAN
  100 DO 120 J = 1,M
      IF (J - K) 105,120,105
  105 W(K,J) = W(K,J)/W(K,K)
      DO 110 I = 1,M
      IF (I-K) 107,110,107
  107 W(I,J)=W(I,J)-W(K,J)*W(I,K)
      CALL OVERFL (KI)
      GO TO (260,110),KI
  110 CONTINUE
      CALL OVERFL (KI)
      GO TO (260,120),KI
  120 CONTINUE
      DET = DET * W(K,K)
      CALL OVERFL (KI)
      GO TO (122,124),KI
  122 DET=0.
  124 DO 130 I = 1,M
      IF (I - K) 125,130,125
  125 W(I,K) = -W(I,K)/W(K,K)
      CALL OVERFL (KI)
      GO TO (260,130),KI
```

```
130 CONTINUE
    W(K,K) = 1.0/W(K,K)
    CALL OVERFL (KI)
    GO TO (260,150),KI
150 CONTINUE
    DO 250 N = 1,M
    DO 210 L = N,M
    IF (LW(L) - N) 210,215,210
210 CONTINUE
215 IF (L - N)  217,230,217
217 DO 220 J = 1,M
    WTRAN = W(L,J)
    W(L,J) = W(N,J)
220 W(N,J) = WTRAN
    LW(L) = LW(N)
230 DO 235 L = N,M
    MEL=M+L
    IF(LW(MEL)-N)235,240,235
235 CONTINUE
240 IF (L - N)  243,250,243
243 DO 245 I=1,M
    WTRAN = W(I,L)
    W(I,L) = W(I,N)
245 W(I,N) = WTRAN
    MAL=M+L
    MAN=M+N
    LW(MAL)=LW(MAN)
250 CONTINUE
    KER = 1
    RETURN
260 KER = 3
    RETURN
    END
```

Program name - BGC 3

Purpose

Fits the von Bertalanffy growth-in-length curve to unequally spaced age groups with unequal sample sizes for separate ages.

Programmer - Patrick K. Tomlinson

Originating organization - California Department of Fish and Game

Approximate storage - 16,000 (CDC 1604)

This program consists of a main program (BGC 3) and a subroutine (EST). It fits equation (1) by least squares when data are of the form (length, age).

$$L_t = \text{length (at age t)} = A + BR^t; \quad 0 < R < 1 \qquad \ldots\ldots (1)$$

The program will process up to 5000 pairs of observed (L_t, t). There are no restrictions, except the data should represent at least four different values of t.

Output is in the von Bertalanffy form, where:

$$A = L_\infty$$
$$R = e^{-K} \quad \text{or} \quad K = -\log_e R$$
$$B = -L_\infty e^{Kt_0} \quad \text{or} \quad t_0 = [\log_e(-B) - \log_e A]/K$$

Method

Given n pairs (L_t, t) and equation (1), the program minimizes the function,

$$Q = \sum^{n} (L_t - A - BR^t)^2 \qquad \ldots\ldots (2)$$

by use of the partial derivatives evaluated near zero. Since $0 < R < 1$, the program uses the following sequence:

1. Set R = 0.0001
2. Solve for A and B using $\partial Q/\partial A$ and $\partial Q/\partial B$
3. Using the values of A, B and R from steps 1 and 2, evaluate $\partial Q/\partial R$
4. Set R = 0.9999 and repeat steps 2 and 3
5. Check the sign of $\partial Q/\partial R$ (.0001) against the sign of $\partial Q/\partial R$ (.9999)
6. If step 5 does not produce different signs, the program tries R = .01 vs R = .99 and R = .1 vs R = .9. If the signs at step 5 are still not different, the data are considered to be unsuitable for equation (1).
7. If the signs at step 5 (or 6) are different, then the sign of $\partial Q/\partial R$ (.5) is compared to the sign of $\partial Q/\partial R$ (.0001). If the signs differ, then R is between .0001 and .5; otherwise, R is between .5 and .9999. As long as the signs differ, the best estimate of R is between the two values of R producing the signs. The interval is then divided by 2 and the process continued until $|(R_i - R_{i+1})| < .0001$.
8. The program now considers the partials to be sufficiently close to zero and it solves for L_∞, K, and t_0.

The output gives values of the expected length at age using equation (1) evaluated at ages selected by the user. The user enters an initial age (t_1) and a Δt. Then $t_{i+1} = t_i + \Delta t$ are the ages utilized. The user also enters the number of ages to be represented.

The pairs (L_t, t) may be read into the program in 2 different ways. The first way assumes no type of ordering or sorting has occurred and each (L_t, t) represents a single fish. The second method allows for frequency distributions and the user provides a triple (L_t, t, m) where m is the number of times (or some weighting factor) the pair (L_t, t) is to be used.

Error messages:

1. TOO MANY ITERATIONS - implies the program is not providing for $|(R_i - R_{i+1})|$ less than .0001.

2. DATA DO NOT FIT - implies the program failed to get by step 6 and the data cannot be fitted by equation (1).

SUBROUTINE EST - solves for A and B where R is given and evaluates $\partial Q/\partial R$ at the given R.

The cards used are defined as sets:

Set #1	Any no. cards	FORMAT (9A8, I8)
	Col. 1	Blank for each card
	Cols. 2-72	Any little information desired. Use as many cards as necessary.
	Cols. 73-80	Blank for each card
Set #2	1 card	FORMAT (9A8, I8)
	Cols. 1-79	Blank
	Col. 80	Punch numeral (1)
Set #3	1 card	FORMAT (2F10.0, 2I4)
	Cols. 1-10	Youngest age for which you wish to compute length for output. Left-justify and punch decimal.
	Cols. 11-20	Increment between ages for which you wish to compute length for output. Left-justify and punch decimal.
	Cols. 21-24	Number of ages for which you wish computed length in output. Right-justify with no decimal.
	Cols. 25-28	Leave blank if data in form of set #4. Punch 1 in col. 28 if data in form of set #5.
Set #4	$1 \leq$ cards ≤ 1250	FORMAT (8F10.0) ** see set #3 cols. 25-28.
	Cols. 1-80	Each card has 8 fields of 10 columns each. Data are punched as pairs (length, age) with 4 pairs per card. Each value is punched left-justified with decimal. If number of pairs is exact multiple of 4, then last card in this set must be blank.
Set #5	$1 \leq$ cards ≤ 5000	FORMAT (2F10.0, I4) ** see set #3 cols. 25-28.
	Cols. 1-10	Punch length left-justified with decimal.
	Cols. 11-20	Punch age left-justified with decimal.
	Cols. 21-24	Punch number of times (frequency or weighting factor) this length-age to be used. Punch one triple per card until all lengths represented. Last card is blank.

** Repeat all cards for additional jobs.

Last card - punch 9 in column 80 to end job run.

Special notes

Storage can be reduced by lowering the arguments of the variables Y, T and T2 in the dimension statements of BGC 3 and subroutine EST. Revised dimensions should be devisible by four. If dimensions are reduced the upper limits of the DO statements 50 + 1 and 51 + 2 must be lowered accordingly.

BGC 3

The $ and multiple replacement statement are used in this program. Their function is explained in Part 1. Input and out statements READ N, list and PRINT N, list are employed and function as explained in Part 1.

FORTRAN II library functions named LOGF and SIGNF appear in BGC 3. The use of LOGF with a FORTRAN IV compiler can be enabled by inserting the following statement function after the dimension statement:

```
LOGF(X) = ALOG(X)
```

The statements involving SIGNF must be changed for other reasons and the connected statements appear in the table below. A number of non-standard 2-branch IF statements appear in BGC 3. The following table indicates the present statement location in terms of the number of statements by which it follows a printed statement number. The revised statement to be substituted for use on all except CDC 1604, 3600 or 3800 computers, is shown adjacent to the statement location.

Statement Location	Revised Statement
60 + 1	IF (IFLAG.EQ.9) GO TO 31
32	IF (IFLAG) 27, 26, 27
33 + 1	IF (ITEST) 51, 50, 51
3 + 2	IF (Y(I)) 2, 4, 2
51 + 4	IF (YY) 56, 4, 56
30 + 2	IF (SIGN(1.,V1).EQ.SIGN(1.,V2)) GO TO 20
10 + 1	IF (IFLAG.GT.15) GO TO 12
15	IF ((R2-R1).LT.0.0001) GO TO 9
17 + 2	IF (SIGN(1.,V1).EQ.SIGN(1.,V3)) GO TO 11

BGC 3

```
      PROGRAM BGC 3
      DIMENSION Y(5000),T(5000),T2(5000),TITLE(9)
   22 PRINT 24
   24 FORMAT (1H1)
   60 READ 23,(TITLE(I),I=1,9),IFLAG
      IF(IFLAG.EQ.9) 31,32
   32 IF(IFLAG) 27,26
   26 PRINT 23,(TITLE(I),I=1,9) $ GOTO 60
   23 FORMAT(9A8,I8)
   27 READ 33,AGE,DELTA,LIMIT,ITEST
   33 FORMAT (2F10.0,2I4)
      IF(ITEST) 51,50
   50 N=0 $ DO 2 J=1,5000,4 $ K=J+3 $ READ 3,(Y(I),T(I),I=J,K)
    3 FORMAT(8F10.0)
      DO 2 I=J,K $ IF(Y(I)) 2,4
    2 N=N+1
      GOTO 4
   51 N=0 $ L=1 $ DO 52 I=1,5000
      READ 53,YY,TT,NN $ IF(YY) 56,4
   56 M=N+NN $ DO 54 J=L,M $ Y(J)=YY $ N=N+1
   54 T(J)=TT $ L=N+1
   52 CONTINUE
   53 FORMAT(2F10.0 ,I4)
    4 KK=0 $ SY=0. $ IFLAG=0
      DO 6 I=1,N $ SY=SY+Y(I)
    6 T2(I)=T(I)*2. $ R1=.0001 $ R2=.9999 $ Z=.0009
   30 CALL EST (R1,T,T2,Y,SY,A,B,V1,N)
      CALL EST (R2,T,T2,Y,SY,A,B,V2,N)
      IF(SIGNF(1.,V1).EQ.SIGNF(1.,V2)) 20,10
   10 IFLAG=IFLAG+1 $ IF(IFLAG.GT.15 ) 12,15
   15 IF((R2-R1).LT.0.0001) 9,17
   17 R3=R1+(R2-R1)/2.
      CALL EST (R3,T,T2,Y,SY,A,B,V3,N)
      IF(SIGNF(1.,V1).EQ.SIGNF(1.,V3)) 11,16
   16 R2=R3 $ V2=V3 $ GOTO 10
   11 R1=R3 $ V1=V3 $ GOTO 10
    9 FK=-LOGF(R3) $TO=(LOGF(-B)-LOGF(A))/FK $ PRINT 13,A,FK,TO,N
   13 FORMAT(1H0,99(1H*)/1H0,12HL(INFINITY)=F12.4,10X,2HK=F8.4,10X,12HT(
     1SUB ZERO)=F8.4,7X,14HNO. FISH AGED=I4)
      PRINT 42
   42 FORMAT(1H0,99(1H*)/1H0,7X3HAGE,14X,6HLENGTH)
      DO 40 I=1,LIMIT $ S=A+(B*(R3**AGE)) $ PRINT 41,AGE,S
   40 AGE=AGE+DELTA $ GOTO 22
   41 FORMAT(1H0,F10.3,10X,F10.3)
   20 KK=KK+1 $ GOTO(28,28,28,29) KK
   28 R1=R1*10. $ R2=R2-Z $ Z=Z*10. $ GOTO 30
   29 PRINT  21 $ GOTO 22
   21 FORMAT(1H0,17HDATA DOES NOT FIT)
   12 PRINT 14 $ GOTO 9
   14 FORMAT (1H0,15HTOO MANY TRIALS)
   31 STOP
      END


      SUBROUTINE EST(R,T,T2,Y,SY,A,B,V,N)
      DIMENSION  Y(5000),T(5000),T2(5000)
      RT=YR=R2T=TR=YTR=TR2T=0. $ FN=N
      DO 5 J=1,N $ RT=RT+R**T(J) $ YR=YR+Y(J)*(R**T(J))
      R2T=R2T+R**T2(J) $ TR=TR+T(J)*(R**T(J))
      YTR=YTR+Y(J)*T(J)*(R**T(J))
    5 TR2T=TR2T+T(J)*(R**T2(J))
      A=((RT*YR)-(SY*R2T))/((RT**2)-(FN*R2T))
      B=((SY*RT)-(FN*YR))/((RT**2)-(FN*R2T))
      V=A*TR-YTR+B*TR2T
      RETURN
      END
```

Program name - BGC 4

Purpose

Estimates the parameters K and L_∞ of the von Bertalanffy growth-in-length curve when only the lengths of individual fish at two points in time are known. This allows the curve to be fitted to tag release and recovery data.

Programmer - Patrick K. Tomlinson

Originating organization - California Department of Fish and Game

Approximate storage - 16,000 (CDC 1604)

This program consists of a main program (BGC 4) and a subroutine (EST). BGC 4 fits equation (1) by least squares when data are of the form (initial length, final length, time elapsed).

$$L_{t+\Delta t} = L_t R^{\Delta t} + A\,(1 - R^{\Delta t}); \quad 0 < R < 1 \qquad \ldots\ldots\ldots (1)$$

L_t = initial length

$L_{t+\Delta t}$ = final length

Δt = time elapsed

Output is in the von Bertalanffy form, where:

$$L_\infty = A$$

$$K = -\log_e R \qquad \ldots\ldots\ldots (2)$$

Method

Given n triples (L_t, $L_{t+\Delta t}$, Δt) and equation (1), the program minimizes the function,

$$Q = \sum^{n} \left[L_{t+\Delta t} - L_t R^{\Delta t} - A\,(1 - R^{\Delta t}) \right] 2 \qquad \ldots\ldots\ldots (3)$$

by use of the partial derivatives evaluated near zero. Since $0 < R < 1$, the program uses the following sequence:

1. Set R = .0001
2. Solve for A using $\partial Q/\partial A$
3. Using the values of A and R from steps 1 and 2, evaluates $\partial Q/\partial R$
4. Set R = .9999 and repeat steps 2 and 3
5. Check the sign of $\partial Q/\partial R$ (.0001) against the sign of $\partial Q/\partial R$ (.9999)
6. If step 5 does not produce different signs, the program tries R = .001 vs R = .999, R = .01 vs R = .99, and R = .1 vs R = .9. If the signs at step 5 are not different for any of these trials, the data are considered to be unsuitable for equation (1).
7. If the signs at step 5 (or 6) are different, then the sign of $\partial Q/\partial R$ (.5) is compared to the sign of $\partial Q/\partial R$ (.0001). If the signs differ, then R is between .0001 and .5; otherwise, R is between .5 and .9999. As long as the signs differ, the best estimate of R is between the two trial values of R producing the different signs. The interval is then divided by 2 and the process continued until $|(R_i - R_{i+1})| < .0001$.
8. The program now considers the partials to be sufficiently close to zero and it solves for L_∞ and K.

The output gives values of the expected length using equation (1) evaluated at an initial length and time lapse selected by the user. The user enters one initial length (L_t) and a time lapse (Δt). The program computes the final lengths ($L_{t+\Delta t}$), ($L_{t+2\Delta t}$), ($L_{t+3\Delta t}$), ..., ($L_{t+m\Delta t}$). The user chooses m also.

The triples are punched on cards, with one triple per card. No provisions are provided for frequency distributions or weighting factors.

The program will handle up to 5000 triples (L_t, $L_{t+\Delta t}$, Δt).

Error messages:

1. TOO MANY ITERATIONS - implies the program is not providing for $|(R_i - R_{i+1})| < .0001$.

2. DATA DO NOT FIT - implies the program failed to get by step 6.

SUBROUTINE EST - solves for A when R is given and evaluates $\partial Q/\partial R$.

The cards used are defined as sets.

Set #1	Any no. cards	FORMAT (9A8, I8)
	Col. 1	Blank for each card
	Cols. 2-72	Any title information desired. Use as many cards as necessary.
	Cols. 73-80	Blank for each card
Set #2	1 card	FORMAT (9A8, I8)
	Cols. 1-79	Blank
	Col. 80	Punch numeral one (1)
Set #3	1 card	FORMAT (2F10.0, I4)
	Cols. 1-10	Shortest length to be used in computing expected lengths for output. Left-justify and punch decimal.
	Cols. 11-20	Time increment between initial length and final length for computing expected values for output. Left-justify with decimal point.
	Cols. 21-24	Number of times you wish to add the time increment. Also, number of expected values in output. Right-justify with no decimal.
Set #4	$1 \leq$ cards ≤ 5000	FORMAT (3F10.0)
	Cols. 1-30	Data are punched as triples (initial length, final length, time elapsed), one triple per card. Initial length is punched left-justified with decimal in columns 1-10. Final length is punched left-justified with decimal in columns 11-20. Time elapsed is punched left-justified with decimal point in columns 21-30.
Set #5	Blank	Card after data.

** Repeat all cards for additional jobs.

Last card - punch 9 in column 80 to end job run.

Special notes

Storage can be reduced by lowering the dimensions of Y, X, and T in BGC 4 and EST to the desired maximum number of observations. The upper limit of the DO at statement 27 + 2 must be adjusted accordingly. Multiple replacement statements and the $ statement separator are employed as explained in Part 1. The statements READ N, list and PRINT N, list read and write on the system standard input and output units.

The statement function LOGF(X) = ALOG(X) must be inserted directly after the dimension statement in BGC 4 to make the LOGF function operable in a FORTRAN IV environment. In addition, the following special statements must be changed except for compilation on CDC 1604, 3600 or 3800 computers.

BGC 4

Statement Location	Revised Statement
60 + 1	IF (IFLAG.EQ.9) GO TO 71
32 + 2	IF (SIGN(1.,V1).EQ.SIGN(1.,V2)) GO TO 8
10 + 1	IF (IFLAG.GT.20) GO TO 12
15	IF ((R2-R1).LT.0.0001) GO TO 9
17 + 2	IF (SIGN(1.,V1).EQ.SIGN(1.,V3)) GO TO 11

Program BGC 3 employs logic similar to BGC 4 and reading the special notes for that program would be advantageous.

BGC 4

```
      PROGRAM BGC4 (INPUT,OUTPUT)
      DIMENSION Y(5000),X(5000),T(5000),TITLE(9)
   33 FORMAT (2F10.0,I4)
    3 FORMAT (3F10.0)
   22 PRINT 24
   24 FORMAT (1H1)
   60 READ 23,(TITLE(I),I=1,9),IFLAG
      IF(IFLAG.EQ.9) 71,72
   72 IF(IFLAG-1) 26,27,27
   26 PRINT 23,(TITLE(I),I=1,9) $ GOTO 60
   23 FORMAT(9A8,I8)
   27 READ 33,AGE,DELTA,LIMIT
      N=IFLAG=0
      DO 2  J=1,5000
      READ 3, X(J),Y(J),T(J)
      IF(Y(J)) 2,4,2
    2 N=N+1
    4 SY=0. $ DO 6 I=1,N
    6 SY=SY+Y(I) $ R1=.0001 $R2=.9999 $ KK=0 $ Z=.0009
   32 CALL EST (R1,Y,X,T,A,V1,N,SY)
      CALL EST (R2,Y,X,T,A,V2,N,SY)
      IF(SIGNF(1.,V1).EQ.SIGNF(1.,V2)) 8,10
   10 IFLAG=IFLAG+1 $ IF(IFLAG.GT.20) 12,15
   15 IF((R2-R1).LT.0.0001) 9,17
   17 R3=R1+(R2-R1)/2.
      CALL EST(R3,Y,X,T,A,V3,N,SY)
      IF(SIGNF(1.,V1).EQ.SIGNF(1.,V3)) 11,16
   16 R2=R3 $ V2=V3 $ GOTO 10
   11 R1=R3 $ V1=V3 $ GOTO 10
    9 V3=-LOGF(R3)
      PRINT 13,A,V3,N
   13 FORMAT(1H0,100(1H*)/1H0,12HL(INFINITY)=F12.4,10X,2HK=F8.4,10X,
     118HNO. FISH MEASURED=I4)
      PRINT 40
   40 FORMAT(1H0,100(1H*)/1H0,7X,14HINITIAL LENGTH 10X,12HTIME ELAPSED,
     1 10X,12HFINAL LENGTH)
      DO 41 I=1,LIMIT $ EL=AGE*R3**DELTA +A*(1.-R3**DELTA)
      PRINT 42,AGE,DELTA,EL
   41 AGE =EL $ GOTO 22
   42 FORMAT(1H0,7X,F14.4,10X,F12.4,10X,F12.4)
    8 KK=KK+1 $ GOTO(30,30,30,31) KK
   30 R1=R1*10. $ R2=R2-Z $ Z=Z*10. $ GOTO 32
   31 PRINT 150
  150 FORMAT (1H1,17HDATA DOES NOT FIT)
      GO TO 22
   12 PRINT 14 $ GOTO 9
   14 FORMAT (1H1,15HTOO MANY TRIALS)
   71 STOP
      END


      SUBROUTINE EST(R,Y,X,T,A,V,N,SY)
      DIMENSION Y(1000),X(1000),T(1000)
         YR=XR2=XR=R2=RT=TXR=TR2=XYR=YTR=XTR2=XTR=TR=0.
      FN=N
      DO 5 I=1,N $ YR=YR+Y(I)*(R**T(I))
      XR2=XR2+X(I)*(R**(2.*T(I))) $ XR=XR+X(I)*(R**T(I))
      R2=R2+R**(2.*T(I)) $ RT=RT+R**T(I)
      TXR=TXR+T(I)*(X(I)**2)*(R**(2.*T(I)))
      TR2=TR2+T(I)*(R**(2.*T(I)))
      XYR=XYR+X(I)*Y(I)*T(I)*(R**T(I))
      YTR=YTR+Y(I)*T(I)*(R**T(I))
      XTR2=XTR2+X(I)*T(I)*(R**(2.*T(I)))
      XTR=XTR+X(I)*T(I)*(R**T(I))
    5 TR=TR+T(I)*(R**T(I))
      A=(SY-YR+XR2-XR)/(FN+R2-2.*RT)
      V=TXR+A*(A*(TR2-TR)+YTR-2.*XTR2+XTR)-XYR
      RETURN
      END
```

Program name - FPOW

Purpose

Estimates the fishing powers of individual vessels relative to a standard vessel and the densities of fish in time-area strata relative to a standard time-area stratum.

Programmer - Catherine L. Berude

Originating organization - California Department of Fish and Game

Approximate storage - 43,000 (CDC 6600)

This program estimates log fishing powers of individual vessels relative to a standard vessel and log fish density for time-area strata relative to a standard stratum using the method described by Robson (1966). The conversion from log relative fishing power and log density to the original scales employs a bias-correcting factor given in Laurent (1963). The bias which is corrected derives from the fact that log catches are assumed to be normally distributed, and it follows that P_i, the estimate of log relative fishing power for the i^{th} vessel, is also normal with expectation P_i. However, the expected value of exp (P_i) does not equal exp (P_i) and consequently the uncorrected estimate is biased.

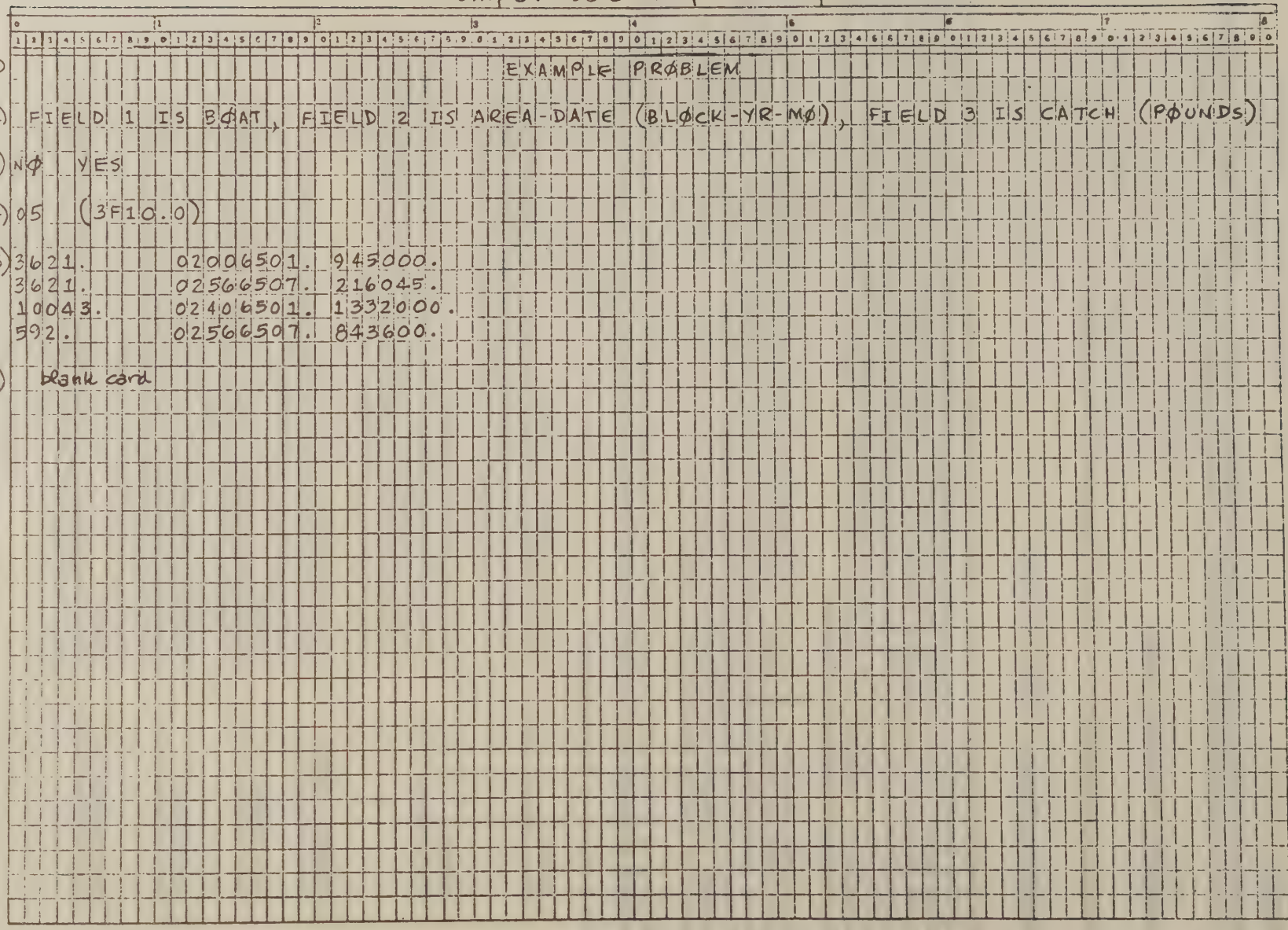
The program handles up to 2000 catch observations from a combined total of not more than 200 distinct boats and time-area strata. Comment cards within the program listing and the data deck set-up instructions assume a constant amount of effort was expended to make each individual catch. If this is not the case, substitute "catch-per-effort" for "catch" each time the latter word appears. Area-date in the instructions means a time-area stratum in which it is assumed the density of fish is constant and uniform.

The program arbitrarily selects the lowest numbered boat as the standard vessel and the lowest numbered area-date in which the standard vessel fished as the standard time-area stratum.

Data deck set-up instructions:

1.	TITLE	One card, Format: (20A4)
	Card columns 1-80	Title information centered on column 40. This title will appear on all output tables; it may identify the source and collection period of the data, date of execution.
2.	SUBTIT	One card, Format: (20A4)
	Card columns 1-80	Title information centered on column 40. Data will appear only in the heading of the table listing observed catch by area-date by boat; it may describe units of catch, area-date selection, boat coding technique.
3.	IVARCOV, JUNK	One card, Format: (A3, 1X, A3)
	Card columns 1-3	Option for printing the variance-covariance matrix. Left-justified, punch YES or NO .
	Card column 4	Blank
	Card columns 5-7	Option for printing a table listing biased estimates and truncation errors. Left-justified, punch YES or NO .
*4.	LUNIT, FMAT	One card, Format: (I2, 2X, 19A4)
	Card columns 1-2	Number of the logical unit through which observed data will be input. For the standard card reader, punch 05; for tape or disk input, punch 20.
	Card columns 3-4	Blank
	Card columns 5-80	The format specification according to which observed data will be read, enclosed in parentheses. Example: (3 F 10.0).

Example: Data Setup for Program FPOW



*5.	C	Data card deck, Format: variable; see card columns 5-80 in 4., above. Observed data in triples of boat number - area-date - catch, punched in the format stated in unit 4.
*6.	End-of-data Card columns 1-80	One card Blank

*Units 4-6 are described for execution of the general subroutine READIN.

Repeat units 1-6 for each additional set of observed data.

Special notes

No storage reduction recommendations are offered for this program. Since the coding is quite complex it will usually be less expensive to locate a large computer rather than attempting to rewrite the program.

Unit 5 is the standard input unit and unit 6 the standard output unit. Subroutine SORTRC writes error messages on unit 61, so that unit number should also be assigned to the standard output unit. Boat, catch, and time-area data are read from a variable input unit as described under 4. in the deck set-up instructions. If these data are on cards they would be read from 5, the standard input unit, and if on tape or disc some other unit would be assigned to LUNIT on card 4. Units 1 and 2 are employed for intermediate storage. It is highly desireable to utilize a drum or disc for units 1 and 2; tapes may be used but they will increase execution time considerably. The statements PRINT N, list and WRITE(6, N) list are both used to write on the standard output unit. A two-branch logical IF statement appears in statement 5+6 of subroutine SYMINV2. For standard FORTRAN IV this should be replaced by:

```
IF (R(J).EQ..FALSE.) GO TO 13
```

Some seven-letter variable names appearing in this program may have to be reduced to six letters. Any FORTRAN IV compiler should give diagnostics for these if they are unacceptable.

FPOW uses substantial amounts of computer time. An idea of total time requirements may be obtained from testing on several test data sets of different sizes with the realization that time usage increases roughly in proportion to the square of the sum of boats plus area-dates.

References

Laurent, Andre G., 1963 Lognormal distribution and the translation method: description and estimation problems. Jour.Amer.Stat.Assn., 58(301):231-235.

Robson, D.S., 1966 Estimation of the relative fishing power of individual ships. Res.Bull.Inter.Comm. NW.Atlant.Fish., (3):5-14.

```
      PROGRAM FPOW (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE61=OUTPUT,
     1TAPE20,TAPE1=4000,TAPE2=4000)
C
C PROGRAM FPOW ESTIMATES RELATIVE FISHING POWER OF INDIVIDUAL VESSELS
C    BY THE METHOD DESCRIBED BY D.S. ROBSON (ICNAF RESEARCH BULL. NO. 3,
C    1966).
C LIMITATIONS:  OBSERVATIONS LESS THAN 2001, DISTINCT BOATS + DISTINCT
C    AREA-DATES LESS THAN 201.  SEVEN DIGITS MAXIMUM FOR EACH BOAT
C    NUMBER, EIGHT MAXIMUM DIGITS FOR EACH AREA-DATE.
C
C FPOW SETS UP OBSERVED DATA, CALLS SUBROUTINES SORTRC AND POWER.
C    POWER PERFORMS FISHING POWER CALCULATIONS AND OUTPUT, REQUIRES
C    SUBROUTINE NVERT AND FUNCTION VECMULT.
C
      DIMENSION  SUBTIT(20),KEY(2)
      COMMON /1/ NAD,AD(200),NB,BN(200)
      COMMON /2/ TITLE(20)
      COMMON /3/ NC,C(2000,3)
      COMMON /4/ Z(20100)
      COMMON /5/ IVARCOV,JUNK
    1 READ (5,800) (TITLE(I),I=1,20),(SUBTIT(I),I=1,20)
  800 FORMAT (20A4)
C
C READ OPTIONS FOR VARIANCE-COVARIANCE TABLE AND BIAS, TRUNC ERR TABLE
C
      READ (5,700) IVARCOV,JUNK
  700 FORMAT (A3,1X,A3)
      DO 10 I=1,2000
      DO 10 J=1,3
      C(I,J) = 0.
   10 CONTINUE
      DO 11 I=1,200
      AD(I) = 0.
      BN(I) = 0.
   11 CONTINUE
C
C DATA INPUT.  OBSVS WITH CATCH GREATER THAN ZERO MUST BE FEWER THAN
C    2001.  NC = NUMBER OF OBSERVATIONS, FIRST COLUMN OF C = BOAT NO.,
C    SECOND COLUMN OF C = AREA-DATE, THIRD COLUMN OF C = CATCH.
      CALL READIN
C
C SORT C ON AREA-DATE, ASCENDING.  CREATE ARRAY, AD, OF DISTINCT
C    AREA-DATES, AND REPLACE ACTUAL AREA-DATES IN C WITH CODED AREA-DTS.
C    NAD = NUMBER OF DISTINCT AREA-DATES.
C
   20 KEY(1) = -2
      CALL SORTRC (KEY,1,3)
      NAD = 1
      ARDT=C(1,2)
      AD(1)=ARDT
      C(1,2) = 1.
      DO 25 I=2,NC
      IF(C(I,2)-ARDT) 21,24,23
   21 WRITE (6,899)
  899 FORMAT (1H1,63H *****  ERROR *****   SUBROUTINE SORTRC WORKING INC
     1ORRECTLY.  EXECUTION TERMINATED.)
      GO TO 100
   23 NAD=NAD+1
      ARDT=C(I,2)
      AD(NAD)=ARDT
   24 C(I,2)=NAD
   25 CONTINUE
```

FPOW

```
C
C SORT C ON AREA-DATE WITHIN BOAT,ASCENDING.  CREATE ARRAY BN OF
C    DISTINCT BOAT NUMBERS.  REPLACE ACTUAL BOAT NOS. IN C WITH CODED
C    BOAT NOS.  NB=NUMBER OF DISTINCT BOATS.
C
      KEY(1) = -1
      KEY(2) = -2
      CALL SORTRC (KEY,2,3)
      NB=1
      BOAT=C(1,1)
      BN(1)=BOAT
      C(1,1)=1.
      DO 30 I=2,NC
      IF(C(I,1)-BOAT) 26,29,27
   26 WRITE (6,899)
      GO TO 100
   27 NB=NB+1
      IF(NAD+NB-202) 28,33,33
   28 BOAT = C(I,1)
      BN(NB)=BOAT
   29 C(I,1)=NB
   30 CONTINUE
      GO TO 40
   33 WRITE (6,898)
  898 FORMAT (1H1,20H *****  ERROR  *****/6X,70H NUMBER OF DISTINCT AREA
     1-DATES PLUS DISTINCT BOAT NUMBERS EXCEEDS 201.//6X,22H EXECUTION T
     2ERMINATED.)
      GO TO 100
C
C PRINTOUT OF SORTED OBSERVED DATA (ACTUAL AREA-DATES AND BOAT NOS.)
C
   40 WRITE (6,850) (TITLE(I),I=1,20),(SUBTIT(I),I=1,20),NC,NB,NAD
  850 FORMAT (1H1,22X,20A4/23X,20A4///50X,26HCATCH BY AREA-DATE BY BOAT/
     1/9X,24HNUMBER OF OBSERVATIONS -,I6,7X,27HNUMBER OF DISTINCT BOATS
     2 -,I4,7X,31HNUMBER OF DISTINCT AREA-DATES -,I4///5X,4HBOAT,6X,
     39HAREA-DATE,6X,5HCATCH,5X,3H***,7X,4HBOAT,6X,9HAREA-DATE,6X,
     45HCATCH,5X,3H***,7X,4HBOAT,6X,9HAREA-DATE,6X,5HCATCH/40X,3H***,
     542X,3H***)
      DO 45 I=1,NC,3
      IB1=C(I,1)
      IB2=C(I+1,1)
      IB3=C(I+2,1)
      IAD1=C(I,2)
      IAD2=C(I+1,2)
      IAD3=C(I+2,2)
      WRITE (6,851) BN(IB1),AD(IAD1),C(I,3),BN(IB2),AD(IAD2),C(I+1,3),
     1BN(IB3),AD(IAD3),C(I+2,3)
  851 FORMAT (3X,F7.0,F13.0,F14.4,2(3X,3H***,F12.0,F13.0,F14.4) )
      ITEMP = I+2
   45 CONTINUE
      IDIF=NC-ITEMP
      IF(IDIF-1) 50,46,47
   46 IB1=C(NC,1)
      IAD1=C(NC,2)
      WRITE (6,852) BN(IB1),AD(IAD2),C(NC,3)
  852 FORMAT (F10.0,F13.0,F14.4,3X,3H***)
      GO TO 50
   47 ITEMP=NC-1
      IB1=C(ITEMP,1)
      IB2=C(NC,1)
      IAD1=C(ITEMP,2)
      IAD2=C(NC,2)
      WRITE (6,853) BN(IB1),AD(IAD1),C(ITEMP,3),BN(IB2),AD(IAD2),C(NC,3)
  853 FORMAT (F10.0,F13.0,F14.4,3X,3H***,F12.0,F13.0,F14.4,3X,3H***)
```

```
C
C IF LOWEST AREA-DATE FOR BOAT 1 IS NOT AREA-DATE 1, INTERCHANGE THOSE
C    AREA-DATES IN ARRAYS C AND AD. (BOAT1,AREA-DATE1 STANDARD)
C
   50 LAD=C(1,2)
      IF(LAD-1) 51,60,51
   51 ADTEMP=AD(1)
      AD(1)=AD(LAD)
      AD(LAD)=ADTEMP
      ARDT=LAD
      C(1,2) = 1.
      DO 55 I=2,NC
      IF(C(I,2)-1.) 53,52,53
   52 C(I,2)=ARDT
      GO TO 55
   53 IF(C(I,2)-ARDT) 55,54,55
   54 C(I,2) = 1.
   55 CONTINUE
      PRINT 5555, (BN(I),I=1,NB),(AD(I),I=2,NAD)
 5555 FORMAT (1H1/(12F11.0/) )
C
C TAKE LNS OF CATCHES
C
   60 DO 65 I=1,NC
      C(I,3) = ALOG(C(I,3))
   65 CONTINUE
C
C  CALL SUBROUTINE POWER,  ARRAYS C, AD, BN, TITLE AND COUNT VARIABLES
C    NC, NAD, NB ARE AVAILABLE TO POWER VIA COMMON.
C
      CALL POWER
      WRITE (6,855)
  855 FORMAT (///// 94H  ***  PROGRAMMED BY BIOMETRICAL ANALYSIS UNIT, M
     1RO, CALIF. DEPT. FISH AND GAME, JANUARY 1969. )
      WRITE (6,856)
  856 FORMAT (1H1)
      GO TO 1
  100 STOP
      END



      SUBROUTINE READIN
C
C THIS SUBROUTINE READIN IS GENERAL, EMPLOYING OPTIONS FOR TAPE OR CARD
C  INPUT (CARDS,LUNIT=5) AND VARIABLE FORMAT SPECIFICATION.  DATA *MUST*
C  BE IN BOAT-AREADATE-CATCH ORDER, NO SORT NECESSARY.
C
C YOU MAY WRITE YOUR OWN.  YOU MUST INCLUDE  COMMON /3/, A COUNT OF THE
C  NUMBER OF OBSERVATIONS (NC), A CHECK FOR ZERO CATCH, AND AN END-OF-
C  OBSERVATIONS CHECK.  WHATEVER THE ORDER YOUR OBS ARE IN, THE C ARRAY
C  UPON RETURN TO FPOW MUST CONTAIN ALL BOAT NUMBERS IN COLUMN 1 -
C  C(*,1),  ALL AREADATES IN COLUMN 2 - C(*,2), AND ALL CATCHES IN
C  COLUMN 3 - C(*,3)
C
      COMMON /3/ NC,C(2000,3)
      DIMENSION  FMAT(19)
      READ (5,1) LUNIT,(FMAT(I),I=1,19)
    1 FORMAT (I2,2X,19A4)
      NC=1
   13 READ (LUNIT,FMAT) (C(NC,J),J=1,3)
```

```
C END-OF-OBSERVATIONS CHECK (AREADATE FIELD = 0)
      IF (C(NC,2)) 18,18,14
C CHECK FOR OBSERVED CATCH OF ZERO.  OBSERVATION DELETED.
   14 IF (C(NC,3)) 13,13,15
C OBSERVATION COUNT CHECK.  IF GR THAN 2000, EXECUTION STOPS.
   15 IF (NC-2000) 16,16,17
   16 NC=NC+1
      GO TO 13
   17 WRITE (6,897)
  897 FORMAT (1H1,87H *****  ERROR  *****   NUMBER OF OBSERVATIONS GREAT
     1ER THAN 2000.  EXECUTION TERMINATED.)
      GO TO 100
   18 NC=NC-1
      RETURN
  100 STOP
      END



      SUBROUTINE POWER
C
C  C CONTAINS CODED AREA-DATES AND BOAT NUMBERS AND LOG CATCH,
C  X IS WORKING AREA FOR 2 COLS OF X MATRIX (MAX DIM 2000 X 200),
C  T1 AND T2 ARE VECTORS USING SAME STORAGE AS X,
C  Y IS VECTOR EQUIV TO AND USED AS LOG CATCHES OF C,
C  REMAINING ARRAYS OCCUPY FIRST 2016 LOCATIONS OF C AND ARE REFERENCED
C    AFTER X MATRIX COMPILED AND WRITTEN ON DISK,
C  BHAT ARE RELATIVE LOG FPS AND ABUNDANCES, ESTB ARE BIASED ESTIMATES
C    OF FPS AND ABUNDANCES, EST ARE UNBIASED ESTIMATES,
C  SEB ARE STANDARD ERRORS OF BHAT, ESTCONF ARE .95 CONFIDENCE LIMITS,
C  RHAT ARE TRUNCATION ERRORS IN CALCULATION OF EST, DIFF ARE BIASES
C    (ESTB - EST)
C
      DIMENSION X(2000,2),T1(2000),T2(2000)
      DIMENSION XPRY(200),BHAT(200),ESTB(200),EST(200),ZDIAG(200),
     1RHAT(200),DIFF(200),SEB(200),ESTCONF(200,2),STORE1(16),Y(2000)
      DIMENSION P(200),Q(200),R(200),LSUB(200),H(2),HH(2)
      COMMON /1/ NAD,AD(200),NB,BN(200)
      COMMON /2/ TITLE(20)
      COMMON /3/ NC,C(2000,3)
      COMMON /4/ Z(20100)
      COMMON /5/ IVARCOV,JUNK
      EQUIVALENCE  (T1(1),X(1,1)),(T2(1),X(1,2))
      EQUIVALENCE  (XPRY(1),C(1,1)),(BHAT(1),C(201,1)),(ESTB(1),C(401,1
     1),(EST(1),C(601,1)),(ZDIAG(1),C(801,1)),(RHAT(1),C(1001,1)),
     2(DIFF(1),C(1201,1)),(SEB(1),C(1401,1)),(ESTCONF(1,1),C(1601,1)),
     3(STORE1(1),C(1,2)),(Y(1),C(1,3))
      EQUIVALENCE (LSUB0,C(199,2)),(LSUB(1),C(200,2)),(P(1),C(400,2)),
     1(Q(1),C(600,2)),(R(1),C(800,2))
      LOGICAL R
      INTEGER ZSUB,ZROW,ROWNO,BEGCOL,ENDCOL,BB,AAD,AD1,AD2,H,HH
      DATA H /6H  BOAT, 9HAREA-DATE /
      DATA HH /1HB,1HA/
      REWIND 1
      REWIND 2
      NDIM = NB+NAD-1
C
C  FOLLOWING STATEMENTS THROUGH 36+3 SET UP X MATRIX TWO COLUMNS AT A
C    TIME, X-TRANSPOSE IS STORED ON DISK AS ROWS, X AS COLUMNS.
C  WITHIN A ROW OF X, BOATS OCCUPY FIRST NB COLUMNS, AREA-DATES
C    EXCLUDING FIRST AREA-DATE OCCUPY THE NEXT NAD-1 COLUMNS.
```

FPOW

```
C
      DO 1 I=1,2000
      DO 1 J=1,2
      X(I,J)=0.
    1 CONTINUE
      DO 2 I=1,20100
      Z(I)=0.
    2 CONTINUE
      DO 10 I=1,NC
      X(I,1)=1.
   10 CONTINUE
      WRITE (1)  (X(I,1),I=1,NC)
      WRITE (2)  (X(I,1),I=1,NC)
      K=1
      NCHK=NB
      IF(NB-2) 36,1111,11
 1111 IEND=1
      LC1=2
      LC2=0
      GO TO 12
   11 IEND=0
      LC1=2
      LC2=3
   12 DO 13 I=1,NC
      DO 13 J=1,2
      X(I,J)=0.
   13 CONTINUE
      DO 20 I=1,NC
      IF(C(I,K)-LC1) 17,16,17
   16 X(I,1)=1.
      GO TO 20
   17 IF(C(I,K)-LC2) 20,18,20
   18 X(I,2)=1.
   20 CONTINUE
      IF(IEND) 25,25,30
   25 WRITE (1) (X(I,1),I=1,NC)
      WRITE (2) (X(I,1),I=1,NC)
      WRITE (1) (X(I,2),I=1,NC)
      WRITE (2) (X(I,2),I=1,NC)
      IF(NCHK-LC2-1) 35,26,27
   26 LC1 = LC2+1
      LC2 = 0
      IEND = 1
      GO TO 12
   27 LC1 = LC2 + 1
      LC2 = LC1 + 1
      GO TO 12
   30 WRITE (1)  (X(I,1),I=1,NC)
      WRITE (2)  (X(I,1),I=1,NC)
   35 IF(K-1) 40,36,40
   36 K=2
      NCHK = NAD
      IF(NAD-2) 37,11,11
   37 IEND=1
      LC1=2
      LC2=0
      GO TO 12
C
C FOLLOWING ROUTINE COMPILES Z MATRIX (ONE ROW OF X-TRANSPOSE X EACH
C     COLUMN OF X = ONE ROW OF Z) AND VECTOR (X-TRANSPOSE X LOG CATCHES)
```

```
C                                       FFOW
   40 REWIND 1
      REWIND 2
      LSUB0=0
      LSUB(1)=NDIM
      DO 1000 KK=2,NDIM
      LSUB(KK)=LSUB(KK-1)+NDIM-KK+1
 1000 CONTINUE
      READ (1) (T1(I),I=1,NC)
      DO 41 I=1,NDIM
      READ (2) (T2(J),J=1,NC)
      Z(I)=VECMULT(T1,T2,NC)
   41 CONTINUE
      XPRY(1)=VECMULT(T1,Y,NC)
      ZSUB=NDIM
      LIM=NDIM-1
      DO 44 NROW=2,LIM
      REWIND 2
      READ (1) (T1(I),I=1,NC)
      J=NROW-1
      DO 42 L=1,J
      READ (2) T2(1)
   42 CONTINUE
      DO 43 L=NROW,NDIM
      READ (2) (T2(I),I=1,NC)
      ZSUB=ZSUB+1
      Z(ZSUB)=VECMULT(T1,T2,NC)
   43 CONTINUE
      XPRY(NROW)=VECMULT(T1,Y,NC)
   44 CONTINUE
      READ (1) (T1(I),I=1,NC)
      ZSUB=ZSUB+1
      Z(ZSUB)=VECMULT(T1,T1,NC)
      XPRY(NDIM)=VECMULT(T1,Y,NC)
 4000 FORMAT (/39F3.0)
C
C INVERSION OF Z
C
      CALL SYMINV2 (NDIM,LSUB,P,Q,R,DET,IERR)
      IF(IERR) 50,50,47
   47 WRITE (6,798)
  798 FORMAT (1H1, 20H *****  ERROR  *****//50H    Z IS A SINGULAR MATRI
     1X.  EXECUTION TERMINATED.)
      GO TO 999
C
C FISHING POWER AND ABUNDANCE CALCULATIONS
C
   50 DO 51 IELEM=1,NDIM
      T1(IELEM)=Z(IELEM)
   51 CONTINUE
      ZDIAG(1)=Z(1)
      ZROW=1
      GO TO 57
   52 ZROW=ZROW+1
      LIM=ZROW-1
      DO 53 IROW=1,LIM
      IELEM=LSUB(IROW-1)+ZROW-IROW+1
      T1(IROW)=Z(IELEM)
   53 CONTINUE
      K=ZROW-1
      LO=LSUB(LIM)+1
      LIM=LSUB(ZROW)
      DO 54 IELEM=LO,LIM
      K=K+1
      T1(K)=Z(IELEM)
```

```
   54 CONTINUE
      ZDIAG(ZROW)=Z(LO)
      GO TO 57
   55 DO 56 IROW=1,NDIM
      IELEM=LSUB(IROW)
      T1(IROW)=Z(IELEM)
      LO=LSUB(NDIM)
      ZDIAG(NDIM)=Z(LO)
   56 CONTINUE
      ZROW=ZROW+1
   57 BHAT(ZROW)=VECMULT(T1,XPRY,NDIM)
      ESTB(ZROW)=EXP(BHAT(ZROW))
      IF(ZROW-NDIM) 52,55,59
   59 CONTINUE
      YPRY = VECMULT(Y,Y,NC)
      BPRXPRY = VECMULT(BHAT,XPRY,NDIM)
      ITEMP = NC-NDIM
      SIGHAT = (YPRY - BPRXPRY) / ITEMP
C CALCULATE RECURSIVE TERMS OF EST EQUATION
      STORE1(1) =-0.5*SIGHAT
      TEMP = ITEMP*SIGHAT
      DO 60 I=2,16
      STORE1(I) =(STORE1(I-1)*TEMP)/(-2*I*(ITEMP+(2*I)-2))
   60 CONTINUE
      DO 70 I=1,NDIM
      TEMP = 0.
      DO 65 J=1,15
      TEMP = TEMP + ( (ZDIAG(I)**J) * STORE1(J) )
   65 CONTINUE
      EST(I) = ESTB(I) * (1. + TEMP)
      SEB(I) = SQRT(SIGHAT*ZDIAG(I))
      ESTCONF(I,1) = EXP(BHAT(I)-1.96*SEB(I))
      ESTCONF(I,2) = EXP(BHAT(I)+1.96*SEB(I))
   70 CONTINUE
C
C PRINTOUT OF FISHING POWERS AND DENSITIES
C
      WRITE (6,701) (TITLE(I),I=1,20),(BN(I),EST(I),SEB(I),
     7(ESTCONF(I,J),J=1,2),I=2,NB)
  701 FORMAT (1H1,22X,20A4///36X,56HESTIMATED RELATIVE FISHING POWERS AN
     1D RELATIVE DENSITIES//44X,8HRELATIVE,9X,14HSTANDARD ERROR,9X,22H0.
     295 CONFIDENCE LIMITS/24X,4HBOAT,13X,13HFISHING POWER,4X,20HOF LOG
     3FISHING POWER,9X,16HON FISHING POWER//(22X,F7.0,F22.3,F21.5,
     3F19.3,F15.3/))
      ISUB=NB+1
      WRITE (6,702) (AD(I-NB+1),EST(I),SEB(I),(ESTCONF(I,J),J=1,2),
     1I=ISUB,NDIM)
  702 FORMAT (///44X,8HRELATIVE,9X,14HSTANDARD ERROR,9X,22H0.95 CONFIDENC
     1E LIMITS/21X,9HAREA-DATE,14X,7HDENSITY,10X,14HOF LOG DENSITY,15X,
     210HON DENSITY//(21X,F8.0,F22.3,F21.5,F19.3,F15.3/))
      WRITE (6,703)  EST(1),BN(1),AD(1)
  703 FORMAT (/// 9H ***  C= ,F10.3,38H   IS EXPECTED CATCH OF STANDARD
     1BOAT ,F7.0,25H   IN STANDARD AREA-DATE ,F8.0,1H.)
C
C OPTIONAL PRINTOUT OF BIAS, TRUNCATION ERROR TABLE
C
      IF(JUNK.EQ.2HNO) GO TO 90
      DO 75 I=1,NDIM
      RHAT(I) = ESTB(I)*( ZDIAG(I)**16 )*STORE1(16)
      DIFF(I) = ESTB(I)-EST(I)
   75 CONTINUE
      IF(NB-NAD) 76,76,78
   76 ITEMP=NB
      GO TO 80
```

```
   78 ITEMP=NAD
   80 WRITE (6,705) (TITLE(I),I=1,20),(BN(I),ESTB(I),DIFF(I),RHAT(I),
     1AD(I),ESTB(I+NB-1),DIFF(I+NB-1),RHAT(I+NB-1),I=2,ITEMP)
  705 FORMAT (1H1,22X,20A4///14X,15HBIASED ESTIMATE,17X,10HTRUNCATION,
     A6X,3H***,17X,15HBIASED ESTIMATE,17X,10HTRUNCATION/5X,4HBOAT,6X,
     B13HFISHING POWER,6X,4HBIAS,11X,5HERROR,8X,3H***,5X,9HAREA-DATE,7X,
     C7HDENSITY,9X,4HBIAS,11X,5HERROR/62X,3H***/(3X,F7.0,F14.3 ,F15.5,
     D   E17.5,6X,3H***,5X,F8.0,F14.3,F15.5,E17.5))
      IF(NB-NAD) 81,85,82
   81 ISUB=2*NB
      WRITE (6,706) (AD(I-NB+1),ESTB(I),DIFF(I),RHAT(I),I=ISUB,NDIM)
  706 FORMAT (62X,3H***,5X,F8.0,F14.3,F15.5,E17.5)
      GO TO 85
   82 ISUB=NAD+1
      WRITE (6,707) (BN(I),ESTB(I),DIFF(I),RHAT(I),I=ISUB ,NB)
  707 FORMAT (3X,F7.0,F14.3,F15.5,E17.5,6X,3H***)
   85 WRITE (6,708) ESTB(1),BN(1),AD(1)
  708 FORMAT (/// 9H ***  C= ,F10.3,45H   IS BIASED EXPECTED CATCH OF ST
     1ANDARD BOAT ,F7.0,25H   IN STANDARD AREA-DATE ,F8.0,1H.)
C
C OPTIONAL PRINTOUT OF VARIANCE-COVARIANCE TABLE
C
   90 IF(IVARCOV.EQ.2HNO) GO TO 100
      ITEMP = NC-NDIM
      CALL VARCOV (NDIM,SIGHAT,ITEMP)
  100 RETURN
  999 STOP
      END
```

```
      FUNCTION VECMULT (A,B,K)
      DIMENSION A(K),B(K)
      VECMULT = 0.
      DO 1 I=1,K
      VECMULT = VECMULT + A(I)*B(I)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VARCOV (NDIM,SIGHAT,ITEMP)
      COMMON /1/ NAD,AD(200),NB,BN(200)
      COMMON /2/ TITLE(20)
      COMMON /3/ IDUM(2199),LSUB0,LSUB(200)
      COMMON /4/ Z(20100)
      INTEGER  BEGCOL,ENDCOL,AD1,AD2,H,HH,ROWNO
      DIMENSION H(2),HH(2)
      DATA H /6H  BOAT,9HAREA-DATE/
      DATA HH /2H(6,2H(A/
      LIM=LSUB(NDIM)
      DO 1 I=1,LIM
      Z(I)=Z(I)*SIGHAT
    1 CONTINUE
      PRINT 7700, (TITLE(I),I=1,20)
```

```
 7700 FORMAT(1H1,24X20A4///51X,25HVARIANCE-COVARIANCE TABLE//10X,108HVAR
     1IANCE-COVARIANCE MATRIX FOLLOWS IN TABULAR FORM.  EACH SECTION LIS
     2TS ELEMENTS OF ONE ROW OF THE SYMMETRIC/10X,108HMATRIX.  ONLY ELEM
     3ENTS OF THE UPPER TRIANGLE ARE PRINTED--THE FIRST VALUE LISTED FOR
     4 EACH SECTION IS AN ELE-/10X,108HMENT OF THE MAIN DIAGONAL.  MATRI
     5X COLUMN NUMBERS IN MULTIPLES OF 8 ELEMENTS PER OUTPUT ROW ARE IND
     6ICATED IN/10X,108HTHE LEFTMOST COLUMN OF OUTPUT.  BOAT OR AREADATE
     7 NUMBERS LABELING ELEMENT VALUES PRINTED ARE PRECEDED BY A B/10X,
     8 54HIF THE LABEL IS A BOAT NUMBER, OR BY AN A IF AREADATE. / )
       IX=0
       AD2=1
       DO 100 ROWNO=1,NDIM
       IF(NB-ROWNO) 5,8,8
     5 J=ROWNO-NB+1
       PRINT 7701, H(2),AD(J),ROWNO
 7701 FORMAT (//2X,6HMATRIX,37X,A10,F8.0,13H (MATRIX ROW ,I3,1H) /
     A1X,7HCOLUMNS )
       GO TO 10
     8 PRINT 7701, H(1),BN(ROWNO),ROWNO
    10 IX=IX+1
       IF(IX-8) 15,15,12
    12 IX=1
    15 NELEM=8
       LO=LSUB(ROWNO-1)+1
       LIM=LO+8-IX
       BEGCOL=ROWNO
       ENDCOL=BEGCOL+8-IX
       IF(ENDCOL-NB) 20,20,35
    20 GO TO (21,22,23,24,25,26,27,28),IX
C BEG PRINTROW BOAT ONLY
    21 PRINT 7702, (HH(1),BN(I),I=BEGCOL,ENDCOL)
 7702 FORMAT (/9X,8(3X,A2,F9.0,1H) ) )
       PRINT 7703, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
 7703 FORMAT (I4,1H-,I3,1X,8E15.5)
       GO TO 80
    22 PRINT 7704, (HH(1),BN(I),I=BEGCOL,ENDCOL)
 7704 FORMAT (/24X,7(3X,A2,F9.0,1H) ) )
       PRINT 7705, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
 7705 FORMAT (I4,1H-,I3,16X,7E15.5)
       GO TO 80
    23 PRINT 7706, (HH(1),BN(I),I=BEGCOL,ENDCOL)
 7706 FORMAT (/39X,6(3X,A2,F9.0,1H) ) )
       PRINT 7707, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
 7707 FORMAT (I4,1H-,I3,31X,6E15.5)
       GO TO 80
    24 PRINT 7708, (HH(1),BN(I),I=BEGCOL,ENDCOL)
 7708 FORMAT (/54X,5(3X,A2,F9.0,1H) ) )
       PRINT 7709, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
 7709 FORMAT (I4,1H-,I3,46X,5E15.5)
       GO TO 80
    25 PRINT 7710, (HH(1),BN(I),I=BEGCOL,ENDCOL)
 7710 FORMAT (/69X,4(3X,A2,F9.0,1H) ) )
       PRINT 7711, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
 7711 FORMAT (I4,1H-,I3,61X,4E15.5)
       GO TO 80
    26 PRINT 7712, (HH(1),BN(I),I=BEGCOL,ENDCOL)
 7712 FORMAT (/84X,3(3X,A2,F9.0,1H) ) )
       PRINT 7713, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
 7713 FORMAT (I4,1H-,I3,76X,3E15.5)
       GO TO 80
    27 PRINT 7714, (HH(1),BN(I),I=BEGCOL,ENDCOL)
```

```
 7714 FORMAT (/99X,2(3X,A2,F9.0,1H) ) )
      PRINT 7715, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
 7715 FORMAT (I4,1H-,I3,91X,2E15.5)
      GO TO 80
   28 PRINT 7716, HH(1),BN(BEGCOL)
 7716 FORMAT (/117X,A2,F9.0,1H) ) )
      PRINT 7717, BEGCOL,Z(LO)
 7717 FORMAT (I8,106X,E15.5)
      GO TO 80
C  BEG PRINTROW SPLIT OR A-DS ONLY
   35 IF(ENDCOL-NDIM) 37,36,36
   36 NELEM=8-ENDCOL+NDIM
      ENDCOL=NDIM
      AD2=NAD
      LIM=LSUB(ROWNO)
   37 IF(NB+8-ENDCOL) 60,40,40
C  SPLIT BEGPRINTROW
   40 AD1=2
      NBE=NB-BEGCOL+1
      IF(NELEM-8) 45,42,42
   42 AD2=10-NBE-IX
   45 IF(NBE) 46,70,50
   46 AD1=AD1-NBE
      GO TO 70
   50 GO TO (51,52,53,54,55,56,57),IX
   51 PRINT 7702, (HH(1),BN(I),I=BEGCOL,NB),(HH(2),AD(I),I=AD1,AD2)
      PRINT 7703, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   52 PRINT 7704, (HH(1),BN(I),I=BEGCOL,NB),(HH(2),AD(I),I=AD1,AD2)
      PRINT 7705, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   53 PRINT 7706, (HH(1),BN(I),I=BEGCOL,NB),(HH(2),AD(I),I=AD1,AD2)
      PRINT 7707, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   54 PRINT 7708, (HH(1),BN(I),I=BEGCOL,NB),(HH(2),AD(I),I=AD1,AD2)
      PRINT 7709, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   55 PRINT 7710, (HH(1),BN(I),I=BEGCOL,NB),(HH(2),AD(I),I=AD1,AD2)
      PRINT 7711, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   56 PRINT 7712, (HH(1),BN(I),I=BEGCOL,NB),(HH(2),AD(I),I=AD1,AD2)
      PRINT 7713, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   57 PRINT 7714, HH(1),BN(NB),HH(2),AD(AD2)
      PRINT 7715, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
C  BEGPRINTROW AREA-DATES ONLY
   60 AD1=BEGCOL-NB+1
      IF(NELEM-8) 70,65,65
   65 AD2=AD1+8-IX
   70 GO TO(71,72,73,74,75,76,77,78),IX
   71 PRINT 7702, (HH(2),AD(I),I=AD1,AD2)
      PRINT 7703, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   72 PRINT 7704, (HH(2),AD(I),I=AD1,AD2)
      PRINT 7705, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   73 PRINT 7706, (HH(2),AD(I),I=AD1,AD2)
      PRINT 7707, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
```

```
   74 PRINT 7708, (HH(2),AD(I),I=AD1,AD2)
      PRINT 7709, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   75 PRINT 7710, (HH(2),AD(I),I=AD1,AD2)
      PRINT 7711, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   76 PRINT 7712, (HH(2),AD(I),I=AD1,AD2)
      PRINT 7713, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   77 PRINT 7714, (HH(2),AD(I),I=AD1,AD2)
      PRINT 7715, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
   78 PRINT 7716, HH(2),AD(AD1)
      PRINT 7717, BEGCOL,Z(LO)
      GO TO 90
C  BOAT MIDPRINTROWS
   80 LO=LIM+1
      LIM=LIM+8
      BEGCOL=ENDCOL+1
      ENDCOL=ENDCOL+8
      IF(ENDCOL-NB) 83,83,85
   83 PRINT 7702, (HH(1),BN(I),I=BEGCOL,ENDCOL)
      PRINT 7703, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 80
C  SPLIT MIDPRINTROW
   85 IF(BEGCOL-NB) 86,86,95
   86 AD1=2
      NBE=NB-BEGCOL+1
      IF(ENDCOL-NDIM) 88,88,87
   87 AD2=NAD
      LIM=LSUB(ROWNO)
      NELEM=8-ENDCOL+NDIM
      ENDCOL=NDIM
      GO TO 89
   88 AD2=9-NBE
   89 PRINT 7702, (HH(1),BN(I),I=BEGCOL,NB),(HH(2),AD(I),I=AD1,AD2)
      PRINT 7703, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
C  A-D MIDPRINTROW
   90 IF(NELEM-8) 100,92,92
   92 LO=LIM+1
      LIM=LIM+8
      BEGCOL=ENDCOL+1
      ENDCOL=ENDCOL+8
   95 AD1=AD2+1
      AD2=AD2+8
      IF(ENDCOL-NDIM) 98,98,96
   96 NELEM=8-ENDCOL+NDIM
      ENDCOL=NDIM
      AD2=NAD
      LIM=LSUB(ROWNO)
   98 PRINT 7702,(HH(2),AD(I),I=AD1,AD2)
      PRINT 7703, BEGCOL,ENDCOL,(Z(I),I=LO,LIM)
      GO TO 90
  100 CONTINUE
      WRITE (6,713) SIGHAT,ITEMP
  713 FORMAT (/////2X,20HRESIDUAL VARIANCE = ,E14.5//2X,21HDEGREES OF FR
     1EEDOM = ,I4)
      RETURN
      END
```

```
      SUBROUTINE  SYMINV2 (N,LOC,P,Q,R,DET,IERR)
      DIMENSION LOC(200),P(200),Q(200),R(200)
      COMMON /4/ A(20100)
      LOGICAL R
      DO 5 I=1,200
      P(I)=0.
      Q(I)=0.
      R(I)=.TRUE.
    5 CONTINUE
      DET=1.
      DO 50 I=1,N
      BIGAJJ=0.
      DO 13 J=1,N
      K=LOC(J-1)+1
      IF(R(J)) 10,13
   10 IF(ABS(A(K))-BIGAJJ) 13,13,11
   11 BIGAJJ=ABS(A(K))
      KSAVE=K
      NPIV=J
   13 CONTINUE
      DET=DET*BIGAJJ
      IF(BIGAJJ) 99,99,16
   16 R(NPIV)=.FALSE.
      P(NPIV)=1.
      Q(NPIV)=1./A(KSAVE)
      A(KSAVE)=0.
      LIM=NPIV-1
      IF(NPIV.LE.1) GO TO 21
      DO 20 IROW=1,LIM
      IELEM=LOC(IROW-1)+NPIV-IROW+1
      P(IROW)=A(IELEM)
      IF(R(IROW)) 17,18
   17 Q(IROW)=-A(IELEM)*Q(NPIV)
      GO TO 19
   18 Q(IROW)=A(IELEM)*Q(NPIV)
   19 A(IELEM)=0.
   20 CONTINUE
   21 IF(NPIV.EQ.N) GO TO 26
      IROW=NPIV
      LO=LOC(LIM)+2
      LIM=LOC(NPIV)
      DO 25 IELEM=LO,LIM
      IROW=IROW+1
      IF(R(IROW)) 22,23
   22 P(IROW)=A(IELEM)
      GO TO 24
   23 P(IROW)=-A(IELEM)
   24 Q(IROW)=-A(IELEM)*Q(NPIV)
      A(IELEM)=0.
   25 CONTINUE
   26 DO 30 IROW=1,N
      PSUBJ=P(IROW)
      K=IROW-1
      LO=LOC(K)+1
      LIM=LOC(IROW)
      DO 29 IELEM=LO,LIM
      K=K+1
      A(IELEM)=A(IELEM)+(PSUBJ*Q(K))
   29 CONTINUE
   30 CONTINUE
   50 CONTINUE
   75 IERR=0
      RETURN
   99 IERR=1
      RETURN
      END
```

```
      SUBROUTINE SORTRC (K,I2,M2)
      DIMENSION K(I2)
      COMMON/3/ N, A(2000,3)
C     PARAMETER DEFINITIONS - A IS ARRAY NAME, M IS ROW DIMENSION OF A
C     (M MUST BE EQUAL TO ROW DIMENSION OF A IN CALLING PROGRAM), LC IS
C     COLUMN DIMENSION OF A AS STATED IN CALLING PROGRAM, K IS SORT KEY
C     VECTOR WHOSE VALUES MUST BE DEFINED IN CALLING PROGRAM, I2 IS
C     NUMBER OF ELEMENTS IN K, M2 IS NUMBER OF LAST COLUMN TO BE SORTED,
C     N IS NUMBER OF ROWS TO BE SORTED.
C     SUBROUTINE SORTS COLUMNS.
C
C     STATEMENTS THROUGH 7 - CHECK VALIDITY OF PARAMETERS.
      IF(M2)300,300,5
    5 IF(N)300,300,6
    6 DO 7 I=1,I2
      IF(K(I))7,301,7
    7 CONTINUE
      N3=N
      N1=N
   11 N1=N1/2
C     CHECK FOR COMPLETION OF SORT.
      IF(N1-1)200,12,12
   12 L1=1
   13 I=1
      K1=IABS(K(I))
      L2=L1+N1
C     CHECK FOR END OF ARRAY.
      IF(L2-N3)14,14,11
   14 IF(K(I))16,301,18
C     ASCENDING SORT (K VALUE NEGATIVE)
   16 IF(A(L1,K1)-A(L2,K1))50,51,52
C     DESCENDING SORT (K VALUE POSITIVE)
   18 IF(A(L1,K1)-A(L2,K1))52,51,50
   50 L1=L1+1
      GO TO 13
   51 I=I+1
      IF(I2-I)50,70,70
   70 K1=IABS(K(I))
      GO TO 14
C     INTERCHANGING SEQUENCE
   52 DO 500 M1=1,M2
      BTEMP = A(L1,M1)
      A(L1,M1) = A(L2,M1)
  500 A(L2,M1) = BTEMP
      L11=L1
C     CHECK FOR BEGINNING OF ARRAY.
   30 IF(L11-N1)50,50,31
C     AFTER INTERCHANGE, CHECK PRECEDING PAIR FOR POSSIBLE REORDERING.
   31 L33=L11-N1
      L22=L11
      L11=L33
      I = 1
      K1=IABS(K(1))
   32 IF(K(I))34,301,35
C     ASCENDING SORT (K VALUE NEGATIVE)
   34 IF(A(L11,K1)-A(L22,K1))50,57,58
C     DESCENDING SORT (K VALUE POSITIVE)
   35 IF(A(L11,K1)-A(L22,K1))58,57,50
C     INTERCHANGING SEQUENCE
   58 DO 600 M1=1,M2
      BTEMP = A(L11,M1)
      A(L11,M1) = A(L22,M1)
```

```
  600 A(L22,M1) = BTEMP
C     CHECK FOR BEGINNING OF ARRAY.
      IF(L11-N1)50,50,37
C     AFTER INTERCHANGE, CHECK PRECEDING PAIR FOR POSSIBLE REORDERING.
   37 L11=L11-N1
      L22=L11+N1
      I=1
      K1=IABS(K(I))
      GO TO 32
C     SORT KEY IS INCREMENTED.  CHECK FOR END OF SORT KEY.
   57 I=I+1
      IF(I2-I)50,33,33
   33 K1=IABS(K(I))
      GO TO 32
  200 RETURN
C     ERROR MESSAGES.
  300 WRITE (61,995)
      STOP
  301 WRITE (61,996)
      STOP
  995 FORMAT (15H N,M2  .ERROR.)
  996 FORMAT (14H K=0   .ERROR.)
      END
```

Program name - MURPHY

Purpose

Calculates the population size in numbers at the beginning of each of a sequence of intervals and the instantaneous fishing mortality coefficient over each interval for a cohort of fish. Assumed known are the catch in numbers during each interval and the instantaneous natural mortality coefficient for each interval.

Programmer - Patrick K. Tomlinson

Originating organization - California Department of Fish and Game.

Approximate storage - 13,000 (CDC 6600)

The method which is programmed here is described by Tomlinson (1970) and his paper should be read before attempting to use this program. References to "text" in the subroutine descriptions refer to his paper. The procedure is generally useful when applied to aged landings data in the case where natural mortality is assumed known and fishing mortality is known for the first or last interval. It also may be used on tagging data when Type 1 tag loss is assumed zero, or when it is known and a correction for it can be made. In the tagging situation the number of tags released is treated as the initial population, the recoveries during time intervals are analogous to catches, and natural mortality is assumed known. Then, a sequence of values can be tried for fishing mortality during the last interval, [F(N)] , until the program estimates an initial population size which is equal to the number of tags released. Here a knowledge of F(N) has been replaced by the assumption that the initial population is known and a complete sequence of fishing mortalities is estimated.

The program handles up to 999 time intervals. This program will solve up to 99 problems, when each problem has a fixed array of ratios [R], a fixed array of catches [C], a fixed array of instantaneous natural mortality [FM], a fixed array of times [T], and a single FF as a starting value for instantaneous fishing mortality. The program has options for:

1. Catches, C(I), to be read in and the R(I) to be computed from the catches. Or, catches to be read in and the R(I) to be computed outside the program and read in. R(I) = C(I+1)/C(I).

2. Times, T(I), to be read in as an array. Or, time read in as a constant, AA, and all T(I) set equal to AA by the program.

3. Natural mortality to be read in as an array. Or, natural mortality read in as a constant, A, and all FM(I) set equal to A.

4. FF read in equal to F(1). Or, FF read in equal to F(N).

Each of the up to 99 problems is treated separately and may utilize any of the options. Conditions are: $C(1) \neq 0$, $C(N) \neq 0$, no 2 consecutive $C(I) = 0$, and $2 \leq N \leq 999$ for each problem. SUBROUTINE SOLVE must be included.

Other types of problems which can be easily programmed involve several trial values for F(1) or F(N) with a fixed array of ratios R(I) or several levels of natural mortality for a fixed array of ratios.

MURPHY

Cards needed for Program MURPHY

Set #1	1 card	FORMAT (I4, F8.0,5I4)
	Cols. 1–4	Value of N = no. of time intervals. Right-justified with no decimal.
	Cols. 5–12	Value of F(1) or F(N). Decimal point should be punched.
	Cols. 13–14	Value of K. Right-justified with no decimal. K = 1 if F(1) known. K = 2 if F(N) known.
	Cols. 15–16	Value of L. Right-justified with no decimal. L = 1 if R(I) to be computed by program. L = 2 if R(I) to be read in by program.
	Cols. 17–18	Value of KK. Right-justified with no decimal. KK = 1 if T(I) is to be read as array. KK = 2 if T(I) is to be read as constant, AA.
	Cols. 19–20	Value of LL. Right-justified with no decimal. LL = 1 if FM(I) is to be read as array. LL = 2 if FM(I) is to be read as constant, A.
	Cols. 21–22	Value of NPROB = no. of problems for this run. This may be left blank for all except the last problem. Right-justified with no decimal.
Set #2	1 ≤ cards ≤ 125	FORMAT (8F10.0)
	Cols. 1–80	Catch data, C(I). Eight catches per card. Each catch is punched in 10 columns with a decimal point. The data are punched in order from C(1) to C(N).
Set #3	1 ≤ cards ≤ 125	FORMAT (8F10.0)
	Optional: Cols. 1–80	If L = 1, these cards omitted. When L = 2. Catch ratios, R(I). Eight ratios per card. Each ratio is punched in 10 columns with a decimal point. The data are punched in order from R(1) to R(N-1).
Set #4	1 ≤ cards ≤ 125	FORMAT (8F10.0)
	Option 1	KK = 1
	Cols. 1–80	Values of the array of time units, T(I). Eight numbers per card. Each number is punched in 10 columns with a decimal point. The data are punched in order from T(1) to T(N).
	Option 2	KK = 2
	Cols. 1–10	Value of AA = constant time for each T(I). Decimal point is punched. The program will set all T(I) equal to AA.
Set #5	1 ≤ cards ≤ 125	FORMAT (8F10.0)
	Option 1	LL = 1
	Cols. 1–80	Values of the array of natural mortality, FM(I). Eight numbers per card. Each number is punched in 10 columns with a decimal point. The data are punched in order from FM(1) to FM(N).
	Option 2	LL = 2
	Cols. 1–10	Value of A = constant natural mortality rate for each FM(I). Decimal point is punched. The program will set all FM(I) equal to A.
Set #6	1 card	FORMAT (20A4)
	Cols. 1–80	Any alphanumeric description of the problem. This will be printed as the heading for the output table.

With reference to the generalized Murphy catch equation, this subprogram solves for E_{i+1} when E_i is known, or for $E_i \exp\left[t_i(F_i + M_i)\right]$ when E_{i+1} is known. It must be called by a main program as discussed previously.

MURPHY

Definition of terms

T = array of time units (t_i in the text)
FM = array of instantaneous natural mortality (M_i in the text)
R = array of catch ratios (R_i in the text)
N = number of time intervals (n in the text)
FF = either F(1) or F(N)
K = 1 if F(1) known; 2 if F(N) known
C = array of catches (such that $C_i = N_i E_i$ of the text)
E = array of exploitation rates (E_i of the text)
POP = array of population size to begin each time interval (N_i of the text)
TOC = total catch ($= \sum^{n} C_i$ in the text)
TOT = total time (= T in the text)
TOF = total instantaneous fishing mortality ($= \sum^{n} t_i F_i$)
TOM = total instantaneous natural mortality ($= \sum^{n} t_i FM_i$)

* FUNCTION FAKE and FUNCTION FISH must be included when using this subroutine.
** No error messages.
*** All arrays are presented as output, plus totals and a title.
**** R(1) and R(N-1) must not equal zero. No 2 consecutive R(I) may be equal to zero.

FUNCTION FAKE

This function subprogram solves (by iteration) for the value of F in the following equation, given that E, FM and T are known, and $0 \leq F \leq 10$.

$$E = F(\exp[T(F+FM)] - 1)/(F+FM)$$

Definition of terms

F = instantaneous fishing mortality
FM = instantaneous natural mortality
E = constant related to exploitation rate
Q1, Q2, Q3 = dummy variables
Y = F at the fifteenth iteration
Z = F at the sixteenth iteration
I = time interval applying to F
T = number time units in I
SIGNF = function which test for sign

* Error message: If (Y-Z)> .0001, a message BAD FIT AT I is printed, program continues with F = Z.
** Accuracy: F ± .0001
*** This function must be included when using SUBROUTINE OPERATE.

MURPHY

FUNCTION FISH

This function subprogram solves (by iteration) for the value of F in the following equation, given E, FM, and T are known, and $0 \leq F \leq 10$.

$$E = F\,(1 - \exp\,[-T(F+M)]/F+FM)$$

Definition of terms

F	=	instantaneous fishing mortality
FM	=	instantaneous natural mortality
E	=	exploitation rate
Q1, Q2, Q3	=	dummy variables
Y	=	F at the fifteenth iteration
Z	=	F at the sixteenth iteration
I	=	time interval to which F belongs
T	=	number time units in I
SIGNF	=	function which test for sign

* Error message: If (Y-Z)> .0001, the message BAD FIT AT I is printed. Program continues with F = Z.

** Accuracy: F ± .0001

*** This function must be included when using SUBROUTINE OPERATE.

Special notes

Storage may be reduced by lowering the dimensions in the main program and all of the subroutines for variables dimensioned 1000. These may be decreased to the maximum number of time intervals desired plus one. READ N, list and PRINT N, list are used to read and write on the standard input and output units.

In subroutines FISH and FAKE the statement functions:

SIGNF(X) = SIGN(X)
EXPF(X) = EXP(X)

should be inserted directly after the dimension statement.

In subroutine SOLVE only the second statement function need be inserted.

Reference

Tomlinson, Patrick K., A generalization of the Murphy catch equation. J.Fish.Res.Bd.Canada, 1970 27:821-825.

MURPHY

```
      PROGRAM MURPHY (INPUT,OUTPUT,TAPE1=INPUT)
      DIMENSION T(1000),FM(1000),R(1000), C(1000),TITLE(20)
C     IF F(1) KNOWN, SET K=1 * * IF C(I) USED FOR R(I), SET L=1
C     IF F(N) KNOWN, SET K=2 * * IF R(I) READ IN,       SET L=2
C     IF TIME VARIABLE FOR THE INTERVALS SET KK=1.
C     IF NATURAL MORTALITY VARIABLE SET LL=1.
C     N=NO. TIME INTERVALS, FF=F(1) OR F(N), NPROB=NO. PROB. SETS.
      IFLAG =0
   12 READ 2,N,FF,K,L,KK,LL,NPROB
    2 FORMAT (I4,F8.0,5I2)
      DO 5 I=1,N,8
      JJ=I+7
    5 READ 6,(C(J),J=I,JJ)
    6 FORMAT(8F10.0)
      IF(L.NE.1) GOTO 4
    3 DO 7 I=2,N
      IF(C(I-1).GT.0.) GOTO 14
   13 R(I-1)=C(I)/C(I-2)
      GOTO 7
   14 R(I-1)=C(I)/C(I-1)
    7 CONTINUE
      GOTO 8
    4 M=N-1
      DO 9 I=1,M,8
      JJ=I+7
    9 READ 6,(R(J),J=I,JJ)
    8 IF(KK.NE.1) GOTO 20
      DO 10 I=1,N,8
      JJ=I+7
   10 READ 6, (T(J),J=I,JJ)
      GOTO 21
   20 READ 6,AA
      DO 22 I=1,N
   22 T(I)=AA
   21 IF(LL.NE.1) GOTO 23
      DO 24 I=1,N,8
      JJ=I+7
   24 READ 6, (FM(J),J=I,JJ)
      GOTO 16
   23 READ 6,A
      DO 18 I=1,N
   18 FM(I)=A
   15 FORMAT(20A4)
   16 READ 15,(TITLE(I),I=1,20)
      CALL SOLVE(T,FM,R,N,FF,K,C,TITLE)
      IFLAG =IFLAG + 1
      IF(IFLAG.NE.NPROB) GOTO 12
   11 STOP
      END



      SUBROUTINE SOLVE(T,FM,R,N,FF,K,C,TITLE)
      DIMENSION T(1000),F(1000),FM(1000),E(1000),R(1000),C(1000),
     1POP(1000),TITLE(20)
      TOC=0.
      TOT=0.
      TOF=0.
      TOM=0.
      IF(K.EQ.2) GOTO 3
```

MURPHY

```
C       THIS ROUTINE ESTIMATES E(I) WHEN F(1)IS KNOWN
      2 E(1)=FF*(1.-EXPF(-T(1)*(FF+FM(1))))/(FF+FM(1))
        DO 4 I=2,N
        IF(E(I-1).LE.0.) GOTO 5
      6 F(I-1)=FISH(E(I-1),FM(I-1),T(I-1),I-1)
        E(I)=R(I-1)*E(I-1)*EXPF(T(I-1)*(F(I-1)+FM(I-1)))
        GOTO 4
      5 F(I-1)=0.
        E(I)=R(I-1)*E(I-2)*EXPF(T(I-2)*(F(I-2)+FM(I-2))+T(I-1)*FM(I-1))
      4 CONTINUE
        F(N)=FISH(E(N),FM(N),T(N),N)
        GOTO 7
C       THIS ROUTINE ESTIMATES E(I) WHEN F(N) IS KNOWN
      3 E(N)=FF*(EXPF(T(N)*(FF+FM(N)))-1.)/(FF+FM(N))
        F(N+1)=FF
        FM(N+1)=FM(N)
        T(N+1)=T(N)
        DO 40 I=2,N
        J=N-I+2
        IF(E(J).LE.0.) GOTO 50
     60 X=(EXPF(T(J+1)*(F(J+1)+FM(J+1)))-1.)/(F(J+1)+FM(J+1))
        F(J)=FAKE(E(J),FM(J),T(J),X,J)
        E(J)=E(J)*EXPF(-T(J)*(F(J)+FM(J)))
        IF(J.EQ.2) GOTO 61
     62 IF (R(J-2).GT.0.) GOTO 61
     51 E(J-1)=0.
        GOTO 40
     61 E(J-1)=E(J)/R(J-1)
        GOTO 40
     50 F(J)=0.
        E(J-1)=E(J+1)*EXPF(-T(J)*FM(J))/R(J)
     40 CONTINUE
        F(1)=FAKE(E(1),FM(1),T(1),X,1)
        E(1)=E(1)*EXPF(-T(1)*(F(1)+FM(1)))
      7 DO 71 I=1,N
        IF (E(I).LE.0.) GOTO 72
     73 POP(I)=C(I)/E(I)
        GOTO 74
     72 POP(I)=POP(I-1)*EXPF(-T(I-1)*(F(I-1)+FM(I-1)))
     74 TOT=TOT+T(I)
        TOF=TOF+T(I)*F(I)
        TOC=TOC+C(I)
     71 TOM=TOM+T(I)*FM(I)
        J=0
        R(N)=0
        L=((N-1)/25)+1
        DO 11 K=1,40
      9 FORMAT(1H1,20A4)
        PRINT 9,(TITLE(I),I=1,20)
        PRINT 12,K,L
        DO 16 I=1,25
        J=J+1
        IF(J.GT.N) GOTO 15
     16 PRINT 14,J,POP(J),C(J),F(J),FM(J),E(J),R(J),T(J)
     11 CONTINUE
     15 PRINT 17,TOC,TOF,TOM,TOT
     17 FORMAT(1H0/1H0,6HTOTALS,13X,F11.1,F13.4,F14.4,24X,F14.2)
     12 FORMAT(1H0,2X,4HTIME,6X,10HPOPULATION,16X,5HINST.,
       59X,5HINST.,7X,8HEXPLOIT.,4X
       1,5HCATCH,7X,9HNUMBER OF/1H ,8HINTERVAL,7X,4HSIZE,7X,5HCATCH,4X,11H
       2FISH. MORT.,4X,10HNAT. MORT.,5X,4HRATE,7X,6HRATIOS,6X,10HTIME UNIT
       3S,3X,4HPAGE,I3,3H OF,I3)
     14 FORMAT(1H0,1X,I4,F14.0,F11.1,F13.4,F14.4,F13.4,F11.4,F14.2)
        RETURN
        END
```

```
      FUNCTION FAKE (E,FM,T,X,I)
      DIMENSION E(1),FM(1),T(1)
      Q1=E
      Y=0.
      Q2=E-(10.*(EXPF(T*(10.+FM))-1.)/(10.+FM))
      Z=10.
      IF(SIGNF(1.,Q1).NE.SIGNF(1.,Q2)) GOTO 2
    1 FAKE=10.
      GOTO 3
    2 DO 4 K=1,16
      FAKE=(Y+Z)/2.
      Q3=E-(FAKE*(EXPF(T*(FAKE+FM))-1.)/(FAKE+FM))
      IF(SIGNF(1.,Q1).NE.SIGNF(1.,Q3)) GOTO 6
    5 Y=FAKE
      Q1=Q3
      GOTO 4
    6 Z=FAKE
    4 CONTINUE
      IF((Y-Z).LE..0001) GOTO 3
    8 PRINT 9,I
    9 FORMAT(1H1,10HBAD FIT AT,I4)
    3 CONTINUE
      RETURN
      END
```

```
      FUNCTION FISH(E,FM,T,I)
      DIMENSION E(1),FM(1),T(1)
      Q1=E
      Y=0.
      Q2=E-(10.*(1.-EXPF(-T*(10.+FM)))/(10.+FM))
      Z=10.
      IF(SIGNF(1.,Q1).NE.SIGNF(1.,Q2)) GOTO 2
    1 FISH=10.
      GOTO 3
    2 DO 4 K=1,16
      FISH=(Y+Z)/2.
      Q3=E-(FISH*(1.-EXPF(-T*(FISH+FM)))/(FISH+FM))
      IF(SIGNF(1.,Q1).NE.SIGNF(1.,Q3)) GOTO 6
    5 Y=FISH
      Q1=Q3
      GOTO 4
    6 Z=FISH
    4 CONTINUE
      IF((Y-Z).LE..0001) GOTO 3
    8 PRINT 9,I
    9 FORMAT(1H1,10HBAD FIT AT,I4)
    3 CONTINUE
      RETURN
      END
```

Program name - FRC 705

ANALYSIS OF A CATCH CURVE

Purpose

To estimate a survival rate from the age composition of a sample.

Programmer - Lawrence E. Gales

Originating organization - Fisheries Research Institute, University of Washington

Approximate storage - 3,000

The general problem of analyzing catch curves is discussed by Chapman and Robson (1960) and Robson and Chapman (1961). This program is based on the results given in these two articles. The user is urged to consult these articles before using the program.

The program computes a number of statistical measures associated with a vector of catch numbers $N_0, N_1, \ldots N_I$ where N_j = number of fish caught of (coded) age "j". Four options are available:

1. OPTION 1 assumes: (1) recruitment and annual survival constant for all age groups entered in catch vector, (2) all ages in catch vector are fully available to sampling gear, (3) ages are known for all fish in catch vector.

   OPTION 1 Computes:
   Estimate of survival rate
   Variance of survival rate
   Standard error of survival rate
   95% confidence interval for survival rate
   Instantaneous mortality rate
   Estimate of instantaneous mortality rate
   Variance of instantaneous mortality rate
   95% confidence interval for mortality rate

2. OPTION 2 tests the hypothesis that the relative frequency in the 0-age group as compared to the older ages does not deviate significantly from the expected frequency under Option 1 assumptions.

   OPTION 2 Computes a chi-square statistic associated with the difference between the best estimate and Heinke's estimate. If this statistic exceeds CHI (a chi-square value for desired confidence level) the catch numbers are recoded as follows:

   $N_1 \dashrightarrow N'_0$
   $N_2 \dashrightarrow N'_1$
   $N_3 \dashrightarrow N'_2$
   .
   .
   .
   $N_I \dashrightarrow N'_{I-1}$

   and the above computations are made for the new vector $N'_0, \ldots N'_{I-1}$.

   This test is repeated until the statistic is less than CHI, a theoretical chi-square value with one degree of degree which specifies the significance level of the test. CHI is entered on a control card. If the statistic is less than CHI, the output is the same as in OPTION 1.

3. OPTION 3 is to be used when assumption (1) and (2) of OPTION 1 hold but it is not possible to age fish whose coded age is greater than "K". OPTION 3 assumes that the recorded relative frequencies are not reliable for fish of ages K+1, K+2, ..., I in the vector of catch numbers.

OPTION 3 Sums the catch for ages K+1 to I and computes the same output as in OPTION 1 using the catch vector N_0, N_1, ... N_K, m where $m = N_{K+1} + \ldots + N_I$.

4. OPTION 4 permits the user to subdivide the catch curve into a number of segments. The assumptions listed under OPTION 1 may be satisfied for the consecutive age groups in one segment but not for age groups in different segments of a catch curve. Because segmentation of a catch curve may be exploratory the program allows the use of overlapping segments, i.e. one age group may appear in more than one segment.

OPTION 4 Computes the same output as OPTION 1, for each segment of the catch vector (i_1, i_2), (i_3, i_4), ... (i_p, i_{p+1}) where, if $i_j = A$ and $i_{j+1} = B$, (i_j, i_{j+1}) means the segment N_A, ..., N_B. The estimate of sj, the survival rate for the jth segment of the catch curve, is obtained by solving equation (19) of Chapman and Robson (1960) by iteration.

INPUT

The input common to all options consists of:

Set 1 A set of cards for the catch vector punched according to FORMAT (9F8.0). The catch numbers N_i are punched (with decimal point) in columns 1–8, 9–16, ..., 65–72.

Set 2 One blank card.

Set 3 One parameter card specifying the option, age at full recruitment (AFR), CHI (OPTION 2, only), K+1 (OPTION 3, only) and identifcation punched according to FORMAT (I1, F2.0, F6.3, I4, 6X, 8A6).

Sets 1 and 2 are identical for all options. The parameter card (SET 3) is punched in the following manner:

The option is designated by a single digit number 1, 2, 3, or 4 punched in column 1.

AFR is a 2-digit right adjusted number punched in columns 2 and 3.

CHI is a 5-digit number (plus a decimal point), punched in columns 4–9.

K+1 is a 4-digit right adjusted number (with no decimal point) punched in columns 10–13.

The identification is any alpha-numerical information punched in columns 20–70.

OPTION 4 requires a fourth set of data which reads in the segments (i_1, i_2), (i_3, i_4) ... etc. according to FORMAT (20I3). i_1 is a 3-digit right adjusted number (with no decimal point) punched in columns 1–3, i_2 is punched in columns 4–6, ..., . i_p (p an odd number, hence the beginning of the segment (i_p, i_{p+1})) is always a 3-digit right adjusted number (with no decimal point) punched in columns 3p–2 to 3p.

INPUT EXAMPLES Sample Data:

$N_0 = 118$	$N_4 = 25$	$N_8 = 10$	CHI = 3.84
$N_1 = 73$	$N_5 = 21$	$N_9 = 1$	K+1 = 4
$N_2 = 36$	$N_6 = 16$	I = 9	$(i_1, i_2) = (0, 3)$
$N_3 = 30$	$N_7 = 14$	AFR = 6	$(i_3, i_4) = (2, 4)$

FRG 705

OPTION 1

Set 1

CARD 1 cols.	N_0 1–8	18	N_1 9–16 73
"	N_2 17–24	36	N_3 25–32 30
"	N_4 33–40	25	N_5 41–48 21
"	N_6 49–56	16	N_7 57–64 14
"	N_8 65–72	10	
CARD 2 cols.	N_9 1–8	1	

Set 2

CARD 3 cols.	Blank card

Set 3

	Option	AFR	IDENTIFICATION
CARD 4 cols.	1	6 3	HAUL 20–70 1

OPTION 2 Cards 1, 2 (Set 1), and 3 (Set 2) the same

Set 3

	Option	AFR	CHI	IDENTIFICATION
CARD 4 cols.	2	6 3	3 4–9 . 84	HAUL 20–70 1

OPTION 3 Cards 1, 2 (Set 1), and 3 (Set 2) the same

Set 3

	Option	AFR	CHI	K+1	IDENTIFICATION
CARD 4 cols.	3	6 3	4–9	10–13 4	HAUL 20–70 1

OPTION 4 Cards 1, 2 (Set 1), and 3 (Set 2) the same

Set 3

	Option	AFR	CHI	K+1	IDENTIFICATION
CARD 4 cols.	4	6 3	4–9	10–13	HAUL 20–70 1

Set 4

	i_1	i_2	i_3	i_4
CARD 5 cols.	1 2 3 0	4 5 6 3	7 8 9 2	10 11 12 4 0

Restrictions

$I \leq 99$

$0 \leq N_i \leq 99999.9$

$0 \leq AFR \leq 99$

Number of segments ≤ 10

Special notes

This program reads from unit 5 and writes on unit 6. An end-of-file check occurs at statement 4+1. The program should operate satisfactorily on non-CDC 6000 computers if this check is removed along with statements 998 and 1999 which follow it. A short compass assembly language subroutine is listed at the end of the FORTRAN. This is the same function subroutine contained in program FRG 708 and reference should be made to the "special notes" section of that program for substitution methods.

References

Chapman, D.C. and D.S. Robson, The analysis of a catch curve. Biometrics, 16:354-368. 1960

Robson, D.S. and D.G. Chapman, Catch curves and mortality rates. Trans.Am.Fish.Soc., 90:181-189. 1961

FRG 705

```
      PROGRAM F705(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C705                                                                  70500020
CFR705                                                                70500030
      LOGICAL BLANK,NONBL
      DIMENSION AN(1000),ISEG(20)                                     70500040
     1,NUM(1000)                                                      70500050
     2,COMNTS(8)                                                      70500060
C     DIMENSION AN(1000),ISEG(20)                                     70500070
C    1,NUM(1000)                                                      70500080
C    2,COMNTS(8)                                                      70500090
 1000 IC=1                                                            70500100
      IE=9                                                            70500110
    4 READ (5,1)(AN(II),II=IC,IE)                                     70500120
      IF(EOF,5) 998,1999
  998 STOP
 1999 CONTINUE
      IF(BLANK(AN(IC)))GO TO 3
    2 IC=IC+9                                                         70500140
      IE=IE+9                                                         70500150
      GO TO 4                                                         70500160
    3 ICC=IC-9                                                        70500170
      DO 5 II=ICC,IC                                                  70500180
      IF(NONBL(AN(II)))GO TO 5
    6 I=II-1                                                          70500200
      GO TO 7                                                         70500210
    5 CONTINUE                                                        70500220
      I=IC-1                                                          70500230
    7 READ (5,8)NOPT,AFR,CHI,KP1,COMNTS                               70500240
      IF (AFR) 700,700,701                                            70500250
700   MFR=-1                                                          70500260
      GO TO 703                                                       70500270
701   CONTINUE                                                        70500280
      MFR=AFR+.00001-1.0                                              70500290
703   CONTINUE                                                        70500300
      DO 710 JI=1,99                                                  70500310
710   NUM(JI)=0                                                       70500320
      DO 240 JI=1,I                                                   70500330
  240 NUM(JI)=JI+MFR                                                  70500340
      GO TO (100,101,102,103),NOPT                                    70500350
  100 WRITE (6,20)AFR,COMNTS                                          70500360
      IC=1                                                            70500370
      IE=9                                                            70500380
  305 IF(BLANK(AN(IC)))GO TO 303
  304 WRITE (6,308)(NUM(IS),IS=IC,IE)                                 70500400
      WRITE (6,309)(AN(II),II=IC,IE)                                  70500410
  302 IC=IC+9                                                         70500420
      IE=IE+9                                                         70500430
      GO TO 305                                                       70500440
  303 CONTINUE                                                        70500450
      AAN=0                                                           70500460
      T=0                                                             70500470
      DO 9 II=1,I                                                     70500480
      T=T+(FLOAT(II-1))*AN(II)                                        70500490
    9 AAN=AAN+AN(II)                                                  70500500
      S=T/(AAN+T-1.0)                                                 70500510
   18 DS2=S*(S-((T-1.0)/(AAN+T-2.0)))                                 70500520
      DS=SQRT(DS2)                                                    70500530
      SCM=S-2.0*DS                                                    70500540
      SCP=S+2.0*DS                                                    70500550
      Z=-ALOG(S)-(((AAN-1.0)*(AAN-2.0))/(AAN*(AAN+T-1.0)*(T+1.0)))    70500560
      VARZ=((1.0-S)**2)/(AAN*S)                                       70500570
```

```
   44 DZ=SQRT(VARZ)                                                    70500580
      ZCM=Z-(2.0*DZ)                                                   70500590
      ZCP=Z+(2.0*DZ)                                                   70500600
      ALNSM=-ALOG(S+(2.0*DS))                                          70500610
      SA=S-(2.0*DS)                                                    70500640
      IF(SA.GT.0.0) ALNSP=-ALOG(SA)
      WRITE (6,21)S,DS2,DS,SCM,SCP,Z,VARZ,DZ,ZCM,ZCP                   70500630
      IF (SA)22,22,23                                                  70500650
   23 WRITE (6,24)ALNSM,ALNSP                                          70500660
      GO TO 1000                                                       70500670
   22 WRITE (6,25)ALNSM                                                70500680
      GO TO 1000                                                       70500690
  101 WRITE (6,30)CHI,AFR,COMNTS                                       70500700
      IC=1                                                             70500710
      IE=9                                                             70500720
  405 IF(BLANK(AN(IC)))GO TO 403
  404 WRITE (6,308)(NUM(IS),IS=IC,IE)                                  70500740
      WRITE (6,309)(AN(II),II=IC,IE)                                   70500750
  402 IC=IC+9                                                          70500760
      IE=IE+9                                                          70500770
      GO TO 405                                                        70500780
  403 CONTINUE                                                         70500790
      IM2=I-2                                                          70500800
      L=0                                                              70500810
   14 T=0                                                              70500820
      AAN=0                                                            70500830
      IL=I-L                                                           70500840
      DO 10 II=1,IL                                                    70500850
      IIK=II+L                                                         70500860
      T=T+(FLOAT(II-1))*AN(IIK)                                        70500870
   10 AAN=AAN+AN(IIK)                                                  70500880
      S=T/(AAN*T-1.0)                                                  70500890
      IT=1+L                                                           70500900
      SH=(AAN-AN(IT))/AAN                                              70500910
      DEN=T*(T-1.0)*(AAN-1.0)                                          70500920
      DENO=AAN*((AAN*T-1.0)**2)*(AAN*T-2.0)                            70500930
      CSS=((S-SH)**2)/(DEN/DENO)                                       70500940
      WRITE (6,33)AFR,S,SH,CSS                                         70500950
      IF(CSS-CHI) 18,12,12                                             70500960
12    BFR=AFR+.00001                                                   70500970
      NFR=BFR                                                          70500980
      IF(NFR-IM2) 11,13,11                                             70500990
   11 AFR=AFR+1.0                                                      70501000
      L=L+1                                                            70501010
      GO TO 14                                                         70501020
   13 WRITE (6,34)                                                     70501030
      GO TO 1000                                                       70501040
  102 WRITE (6,40)AFR,COMNTS                                           70501050
      IC=1                                                             70501060
      IE=9                                                             70501070
505   IF(BLANK(AN(IC))) GO TO 503
  504 WRITE (6,308)(NUM(IS),IS=IC,IE)                                  70501090
      WRITE (6,309)(AN(II),II=IC,IE)                                   70501100
  502 IC=IC+9                                                          70501110
      IE=IE+9                                                          70501120
      GO TO 505                                                        70501130
  503 CONTINUE                                                         70501140
      T=0                                                              70501150
      AAN=0                                                            70501160
      AM=0                                                             70501170
      KP2=KP1+1                                                        70501180
      DO 41 II=KP2,I                                                   70501190
```

FRG 705

```
 41 AM=AM+AN(II)                                                    70501200
    DO 42 II=1,KP1                                                  70501210
    T=T+(FLOAT(II-1))*AN(II)                                        70501220
 42 AAN=AAN+AN(II)                                                  70501230
    T=T+AM*(FLOAT(KP1))                                             70501240
    AAN=AAN+AM                                                      70501250
    S=T/(AAN-AM+T)                                                  70501260
    X=S**KP1                                                        70501270
    DS2=S*((1.0-S)**2)/(AAN*(1.0-X))                                70501280
    DS=SQRT(DS2)                                                    70501290
    SCM=S-(2.0*DS)                                                  70501300
    SCP=S+(2.0*DS)                                                  70501310
    VARZ=((1.0-S)**2)/(AAN*S*(1.0-X))                               70501320
    Z=-ALOG(S)-VARZ                                                 70501330
    WRITE (6,45)                                                    70501340
    WRITE (6,46)(AN(II),II=1,KP1),AM                                70501350
    GO TO 44                                                        70501360
103 READ (5,50)(ISEG(JJ),JJ=1,20)                                   70501370
    WRITE (6,90)COMNTS                                              70501380
    IC=1                                                            70501390
    IE=9                                                            70501400
605 IF(BLANK(AN(IC))) GO TO 603
604 WRITE (6,308)(NUM(IS),IS=IC,IE)                                 70501420
    WRITE (6,309)(AN(II),II=IC,IE)                                  70501430
602 IC=IC+9                                                         70501440
    IE=IE+9                                                         70501450
    GO TO 605                                                       70501460
603 CONTINUE                                                        70501470
    DO 52 JJ=1,19,2                                                 70501480
    IF (ISEG(JJ))53,53,52                                           70501490
 53 IF(ISEG(JJ+1)) 54,54,52                                         70501500
 54 K=(JJ-1)/2                                                      70501510
    GO TO 55                                                        70501520
 52 CONTINUE                                                        70501530
    K=10                                                            70501540
 55 CONTINUE                                                        70501550
    KK=(2*K)-1                                                      70501560
    DO 70 L=1,KK,2                                                  70501570
    NFR=AFR+.0001                                                   70501580
    T=0                                                             70501590
    AAN=0                                                           70501600
    NSEG1=NFR+ISEG(L)                                               70501610
    NSEG2=NFR+ISEG(L+1)                                             70501620
    I1=ISEG(L)+1                                                    70501630
    I2=ISEG(L+1)+1                                                  70501640
    M=(L/2)+(L-(2*(L/2)))                                           70501650
    WRITE (6,57)M,ISEG(L),ISEG(L+1),NSEG1,NSEG2                     70501660
    WRITE (6,232)                                                   70501670
    DO 56 II=I1,I2                                                  70501680
    IT=II-I1                                                        70501690
    T=T+(FLOAT(IT))*AN(II)                                          70501700
 56 AAN=AAN+AN(II)                                                  70501710
    XBAR=T/AAN                                                      70501720
    J=I2-I1                                                         70501730
    S=(AAN-AN(I1))/AAN                                              70501740
    ZJP1=J+1                                                        70501750
    ZJM1=J-1                                                        70501760
    JP1=J+1                                                         70501770
    JM1=J-1                                                         70501780
    AJ2=ZJP1**2                                                     70501790
    LLL=0                                                           70501800
```

```
   63 SJP1=S**JP1                                                    70501810
      LLL=LLL+1                                                      70501820
      SJM1=S**JM1                                                    70501830
      SUM=0                                                          70501840
      SS=0                                                           70501850
      JS=J+1                                                         70501860
      DO 60 JJ=1,JS                                                  70501870
      JL=JJ-1                                                        70501880
      JK=JJ-2                                                        70501890
      SUM=SUM+((FLOAT(JL))*(S**JK))                                  70501900
   60 SS=SS+S**JL                                                    70501910
      SSS=AAN*(SUM/SS)                                               70501920
      DA=(1.0-S)**2                                                  70501930
      DB=1.0/(S*DA)                                                  70501940
      DC=AJ2*SJM1                                                    70501950
      DE=(1.0-SJP1)**2                                               70501960
      DG=DC/DE                                                       70501970
      DD=DB-DG                                                       70501980
      DP=1.0/AAN                                                     70501990
      S=S+((T/S-SSS)*(DP/DD))                                        70502000
      Y=1.0-S                                                        70502010
      U=S/Y                                                          70502020
      XX=XBAR-U+((ZJP1)*(SJP1/(1.0-SJP1)))                           70502030
      XX=ABS(XX)                                                     70502040
      WRITE (6,72)LLL,S,XBAR,XX                                      70502050
      IF (LLL-100) 221,220,220                                       70502060
220   WRITE (6,230)                                                  70502070
      GO TO 1000                                                     70502080
221   CONTINUE                                                       70502090
      IF(XX-.000001)69,63,63                                         70502100
   69 SJP1=S**JP1                                                    70502110
      SJM1=S**JM1                                                    70502120
      AJ2=ZJP1**2                                                    70502130
      DA=(1.0-S)**2                                                  70502140
      DB=1.0/(S*DA)                                                  70502150
      DC=AJ2*SJM1                                                    70502160
      DE=(1.0-SJP1)**2                                               70502170
      DG=DC/DE                                                       70502180
      DD=DB-DG                                                       70502190
      DS2=(1.0/AAN)/DD                                               70502200
      DS=SQRT(DS2)                                                   70502210
      SCM=S-(2.0*DS)                                                 70502220
      SCP=S+(2.0*DS)                                                 70502230
      Z=-ALOG(S)-(DS2/(S**2))                                        70502240
      VARZ=DS2/(S**2)                                                70502250
      DZ=SQRT(VARZ)                                                  70502260
      ZCM=Z-(2.0*DZ)                                                 70502270
      ZCP=Z+(2.0*DZ)                                                 70502280
      WRITE(6,999) S,DS2,DS,SCM,SCP
      ALNSM=-ALOG(S+(2.0*DS))                                        70502290
      WRITE(6,999) S,DS2,DS,SCM,SCP,ALNSM
      SA=S-(2.0*DS)                                                  70502320
      IF(SA.GT.0.0) ALNSP=-ALOG(SA)
999   FORMAT(1H0,6F12.9)
      WRITE (6,21)S,DS2,DS,SCM,SCP,Z,VARZ,DZ,ZCM,ZCP                 70502310
      IF(SA)82,82,83                                                 70502330
   83 WRITE (6,24)ALNSM,ALNSP                                        70502340
      WRITE (6,98)                                                   70502350
      GO TO 70                                                       70502360
   82 WRITE (6,25)ALNSM                                              70502370
      WRITE (6,98)                                                   70502380
   70 CONTINUE                                                       70502390
      GO TO 1000                                                     70502400
```

FRG 705

```
   98 FORMAT(1H ,50(2H--))                                                  70502410
    8 FORMAT(I1,F2.0,F6.3,I4,6X,8A6)                                        70502420
    1 FORMAT(9F8.0)                                                         70502430
230   FORMAT(1H //26H PROCESS DOES NOT CONVERGE)                           70502440
232   FORMAT(1H //1H )                                                     70502450
   57 FORMAT(1H0/1H0,50(2H--)/9H SEGMENT ,I2,5X,2HN(,I3,9H)  TO  N(,I3,1 70502460
     18H)   INCLUDES AGES ,I4,1X,4H TO ,I4)                               70502470
   90 FORMAT(1H1,55(2H**)/15H O P T I O N  4,3X,46HSURVIVAL ESTIMATES FO 70502480
     1R A SEGMENTED CATCH CURVE,6X,8A6/1H ,55(2H**))                      70502490
50    FORMAT(20I3)                                                        70502500
   40 FORMAT(1H1,55(2H**)/15H O P T I O N  3,3X,21H OLDER AGES COMBINED. 70502510
     1,2X,F3.0,32H IS YOUNGEST AGE FULLY AVAILABLE,6X,8A6/1H 55(2H**))    70502520
   46 FORMAT(1H ,9(2X,F10.2))                                             70502530
   45 FORMAT(1H0,19HCOMBINED AGE VECTOR)                                  70502540
   30 FORMAT(1H1,55(2H**)/43H O P T I O N  2  USING CHI SQUARE VALUE OF  70502550
     1,F7.3,9H AND AFR=F4.1,6X,8A6/1H 55(2H**))
   33 FORMAT(1H ////28H AGE AT FULL RECRUITMENT IS F3.0//29H ESTIMATE OF 70502570
     1 SURVIVAL RATE = F11.8/30H HEINCKES SURVIVAL ESTIMATE = F11.8/24H
     2CHI-SQUARE STATISTIC = F10.4)                                       70502590
   34 FORMAT(1H0/1H0,70H ORIGINAL ASSUMPTION OF CONSTANT SURVIVAL AND RE 70502600
     1CRUITMENT NOT TENABLE )                                             70502610
  308 FORMAT(1H0/1H0,4X,3HAGE,5X,9(6X,I4))                                70502620
  309 FORMAT(1H0,4X,6HNUMBER,2X,9(2X,F8.1))                               70502630
   20 FORMAT(1H1,55(2H**)/15H O P T I O N  1,3X,F3.0,32H IS YOUNGEST AGE 70502640
     1 FULLY AVAILABLE,6X,8A6/1H ,55(2H**))                               70502650
   21 FORMAT(2H ////29H ESTIMATE OF SURVIVAL RATE = F12.9/29H VARIANCE O
     1F SURVIVAL RATE = F12.9/35H STANDARD ERROR OF SURVIVAL RATE =  F12
     2.9 /53H 95 PER CENT CONFIDENCE INTERVAL FOR SURVIVAL RATE = F11.8,
     32X,F11.8/32H INSTANTANEOUS MORTALITY RATE = F10.6/44H VARIANCE OF
     4INSTANTANEOUS MORTALITY RATE = F11.8/23H STANDARD ERROR OF Z = F11
     5.8/41H 95 PER CENT CONFIDENCE INTERVAL FOR Z = F10.6,2X,F10.6)
   24 FORMAT(39H Z INTERVAL OBTAINED FROM S INTERVAL = F11.8,2X,F11.8)
   25 FORMAT(39H Z INTERVAL OBTAINED FROM S INTERVAL = F11.8,2X,13HPLUS
     1INFINITY)                                                           70502740
   72 FORMAT(1H ,15HITERATION NO. (I4,6H)  S= ,E14.6,3X,3HX= ,E14.6,3X,7 70502750
     1HERROR= E14.6)                                                      70502760
      END                                                                 70502770
```

```
      IDENT  BLANDK
*     CDC VERSION    10/9/69
*     BLANK AND NONBL ARE LOGIGAL FUNCTIONS
*     TO DISTINGUISH BLANKS IN I OR F FORMAT
*
      ENTRY  BLANK
      ENTRY  NONBL
      VFD    42/0HBLANK,18/1
BLANK BSS    1
      SA1    B1
      PL     X1,FALSE
      NZ     X1,FALSE
TRUE  MX6    60
      JP     BLANK
      VFD    42/0HNONBL,18/1
NONBL BSS    1
      SA1    B1
      SA2    NONBL
      BX6    X2
      SA6    BLANK
      PL     X1,TRUE
      NZ     X1,TRUE
FALSE SX6    B0
      JP     BLANK
      END
```

A blank card must follow card 2+K if and only if m is an exact multiple of 10 but not of 50 (that is, m = 10, 20, 30 and 40). Following this set of K cards are your L cards for the vector $y = (y_1, y_2, \ldots, y_n)$ punched according to FORMAT (10F8.0), i.e.

Card	j	$y_1, y_2, \ldots, y_{10}$
	.	
	.	
Card	j - 1 + L	$y_k, y_{k+1}, \ldots, y_n$. $k = 10(L-1) + 1$

The same conditions on m hold for n.

Special notes

This program reads input from unit 5 and writes on unit 6. An end-of-file check is made at statement 606 + 1. If this statement is removed the program will operate correctly on most computer systems.

Reference

Paulik, G.J. and L.E. Gales, Allometric growth and Beverton and Holt yield equation. 1964 Amer.Fish.Soc.,Trans., 93(4):369-381.

FRG 701

```
      PROGRAM FR701(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C701                                                                  70100020
      DIMENSION X1(4),X2(4),X3(4),X4(4),XX1(4),XX2(4),XX3(4),XX4(4),  70100030
     1F1(4),F2(4),F3(4),F4(4),FF1(4),FF2(4),FF3(4),FF4(4),            70100040
     2Y1(4),Y2(4),Y3(4),Y4(4),YY1(4),YY2(4),YY3(4),YY4(4),XY1(4),XY2(4),70100050
     3XY3(4),XY4(4),F1234(4),FY123(4),FXY12(4),FYY12(4),AREA(50,50)   70100060
     4,PP(20),XZ(20),FA(61),TPP(61)
      COMMON SAVE(50,50),COMP1(50,50),COMP2(50,50),YW(50,50)
C     DIMENSION X1(4),X2(4),X3(4),X4(4),XX1(4),XX2(4),XX3(4),XX4(4),  70100100
C     1F1(4),F2(4),F3(4),F4(4),FF1(4),FF2(4),FF3(4),FF4(4),           70100110
C     2Y1(4),Y2(4),Y3(4),Y4(4),YY1(4),YY2(4),YY3(4),YY4(4),XY1(4),XY2(4),70100120
C     3XY3(4),XY4(4),F1234(4),FY123(4),FXY12(4),FYY12(4),AREA(50,50)  70100130
C     4,PP(20),XZ(20)                                                 70100140
C     5,SAVE(50,50),COMP1(50,50),COMP2(50,50),FA(61),TPP(61)          70100150
C     6,YW(50,50)                                                     70100160
      DO 1 I=1,13                                                     70100170
1     XZ(I)=(.083333333)*(FLOAT(I-1))                                 70100180
      X1(1)=0                                                         70100190
      X2(1)=XZ(2)                                                     70100200
      X3(1)=XZ(3)                                                     70100210
      X4(1)=XZ(4)                                                     70100220
      X1(2)=XZ(4)                                                     70100230
      X2(2)=XZ(5)                                                     70100240
      X3(2)=XZ(6)                                                     70100250
      X4(2)=XZ(7)                                                     70100260
      X1(3)=XZ(7)                                                     70100270
      X2(3)=XZ(8)                                                     70100280
      X3(3)=XZ(9)                                                     70100290
      X4(3)=XZ(10)                                                    70100300
      X1(4)=XZ(10)                                                    70100310
      X2(4)=XZ(11)                                                    70100320
      X3(4)=XZ(12)                                                    70100330
      X4(4)=XZ(13)                                                    70100340
      X4(4)=1.0                                                       70100350
      DO 47 I=1,4                                                     70100360
      XX1(I)=(X1(I)-X2(I))*(X1(I)-X3(I))*(X1(I)-X4(I))                70100370
      XX2(I)=(X2(I)-X1(I))*(X2(I)-X3(I))*(X2(I)-X4(I))                70100380
      XX3(I)=(X3(I)-X1(I))*(X3(I)-X2(I))*(X3(I)-X4(I))                70100390
      XX4(I)=(X4(I)-X1(I))*(X4(I)-X2(I))*(X4(I)-X3(I))                70100400
      Y1(I)=X2(I)+X3(I)+X4(I)                                         70100410
      Y2(I)=X1(I)+X3(I)+X4(I)                                         70100420
      Y3(I)=X1(I)+X2(I)+X4(I)                                         70100430
      Y4(I)=X1(I)+X2(I)+X3(I)                                         70100440
      YY1(I)=X2(I)*X3(I)+(X4(I)*(X2(I)+X3(I)))                        70100450
      YY2(I)=X1(I)*X3(I)+(X4(I)*(X1(I)+X3(I)))                        70100460
      YY3(I)=X1(I)*X2(I)+(X4(I)*(X1(I)+X2(I)))                        70100470
      YY4(I)=X1(I)*X2(I)+(X3(I)*(X1(I)+X2(I)))                        70100480
      XY1(I)=X2(I)*X3(I)*X4(I)                                        70100490
      XY2(I)=X1(I)*X3(I)*X4(I)                                        70100500
      XY3(I)=X1(I)*X2(I)*X4(I)                                        70100510
47    XY4(I)=X1(I)*X2(I)*X3(I)                                        70100520
606   READ (5,200)FO,TPPO,RANGE,UPPER,FDELT,DELTAT,DELTA,DDELTA       70100530
      IF (EOF,5)9099,9001
9001  READ (5,200)R,W,AM,AK,TP,TO,TLAMRA,XINTO
      IF (EOF,5)9099,9002
9002  JAC=1
      WRITE (6,8000)                                                  70100560
8000  FORMAT(1H1//1H 22(5H*****)/1H 10HINPUT DATA/1H 22(5H*****))     70100570
      WRITE (6,8001)                                                  70100580
8001  FORMAT(1H ////5X2HFO,11X4HTPPO,9X5HRANGE,8X5HUPPER,8X5HFDELT,8X6HD70100590
     1ELTAT,7X5HDELTA,8X6HDDELTA)                                     70100600
      WRITE (6,277)FO,TPPO,RANGE,UPPER,FDELT,DELTAT,DELTA,DDELTA      70100610
277   FORMAT(1H 8(2XE11.4))                                           70100620
      WRITE (6,8002)                                                  70100630
```

```
8002  FORMAT(1H //5X1HR,12X1HW,12X2HAM,11X2HAK,11X2HTP,11X2HTO,11X6HTLAM70100640
     18A,7X5HXINTO)                                                    70100650
      WRITE (6,277)R,W,AM,AK,TP,TO,TLAMBA,XINTO                        70100660
      XINT=EXP(-AK*(TLAMBA-TO))                                        70100670
      IF (XINT) 2000,2000,2001                                         70100680
2000  KAT=1                                                            70100690
      GO TO 2002                                                       70100700
2001  KAT=2                                                            70100710
2002  D=DELTA                                                          70100720
200   FORMAT(10F8.0)                                                   70100730
      IST=0                                                            70100740
      N=((RANGE-FO)/FDELT)+(FDELT/6.0)+1.0                             70100750
      M=((UPPER-TPPO)/DELTAT)+(DELTAT/6.0)+1.0                         70100760
      RW=R*W                                                           70100770
      DO 17 I=1,N                                                      70100780
      FA(I)=FO+(FLOAT(I-1))*FDELT                                      70100790
      FRW=FA(I)*RW                                                     70100800
      FM=-FA(I)-AM                                                     70100810
      FMTO=FM*TO                                                       70100820
      DO 17 J=1,M                                                      70100830
      TPP(J)=TPPO+(FLOAT(J-1))*DELTAT                                  70100840
      YW(I,J)=FRW*(EXP(-AM*(TPP(J)-TP)))*(EXP(FMTO-FM*TPP(J)))         70100850
17    YW(I,J)=YW(I,J)/AK                                               70100860
      IF (R) 203,3008,203                                              70100870
3008  DO 18 I=1,N                                                      70100880
      DO 18 J=1,M                                                      70100890
18    YW(I,J)=1.0                                                      70100900
      IF (W) 203,3009,203                                              70100910
3009  IF (XINTO) 4100,4101,4100                                        70100920
4101  XINT=0                                                           70100930
      KAT=1                                                            70100940
      GO TO 607                                                        70100950
4100  XINT=XINTO                                                       70100960
607   IF (FDELT) 3024,3010,3024                                        70100970
3010  KS=1                                                             70100980
      KIS=10                                                           70100990
3013  READ (5,200)(FA(I),I=KS,KIS)                                     70101000
      IF (EOF,5)9099,9093
 9093 IF (FA(KIS)) 3012,3011,3012
3012  KS=KS+10                                                         70101020
      KIS=KIS+10                                                       70101030
      IF (KIS-61) 3013,3011,3011                                       70101040
3011  DO 3015 I=1,51                                                   70101050
      IF (FA(I)) 3015,3014,3015                                        70101060
3014  N=I-1                                                            70101070
      GO TO 3024                                                       70101080
3015  CONTINUE                                                         70101090
      N=50                                                             70101100
3024  JAC=2                                                            70101110
      IF (DELTAT) 203,3030,203                                         70101120
3030  KS=1                                                             70101130
      KIS=10                                                           70101140
3033  READ (5,200)(TPP(I),I=KS,KIS)                                    70101150
      IF (EOF,5)9099,9094
 9094 IF (TPP(KIS)) 3032,3031,3032
3032  KS=KS+10                                                         70101170
      KIS=KIS+10                                                       70101180
      IF (KIS-61) 3033,3031,3031                                       70101190
3031  DO 3035 I=1,51                                                   70101200
      IF (TPP(I)) 3035,3034,3035                                       70101210
3034  M=I-1                                                            70101220
      GO TO 203                                                        70101230
3035  CONTINUE                                                         70101240
      M=50                                                             70101250
```

FRG 701

```
203   DO 10 I=1,4                                                          70101260
      FF1(I)=(1.0-X1(I))**DELTA                                            70101270
      FF2(I)=(1.0-X2(I))**DELTA                                            70101280
      FF3(I)=(1.0-X3(I))**DELTA                                            70101290
      FF4(I)=(1.0-X4(I))**DELTA                                            70101300
      F1(I)=FF1(I)/XX1(I)                                                  70101310
      F2(I)=FF2(I)/XX2(I)                                                  70101320
      F3(I)=FF3(I)/XX3(I)                                                  70101330
      F4(I)=FF4(I)/XX4(I)                                                  70101340
      F1234(I)=F1(I)+F2(I)+F3(I)+F4(I)                                     70101350
      FY123(I)=(F1(I)*Y1(I))+(F2(I)*Y2(I))+(F3(I)*Y3(I))+(F4(I)*Y4(I))     70101360
      FYY12(I)=(F1(I)*YY1(I))+(F2(I)*YY2(I))+(F3(I)*YY3(I))+(F4(I)*YY4(I70101370
     1))                                                                   70101380
10    FXY12(I)=(F1(I)*XY1(I))+(F2(I)*XY2(I))+(F3(I)*XY3(I))+(F4(I)*XY4(I70101390
     1))                                                                   70101400
      GO TO (403,9003),JAC                                                 70101410
9003  DO 9004 I=1,N                                                        70101420
      DO 9004 J=1,M                                                        70101430
9004  YW(I,J)=1.0                                                          70101440
403   DO 93 JK=1,N                                                         70101450
      GO TO (4001,4002),JAC                                                70101460
4002  P=FA(JK)                                                             70101470
      GO TO 4003                                                           70101480
4001  P=(FA(JK)+AM-AK)/AK                                                  70101490
4003  DO 80 IL=1,M                                                         70101500
      KATY=KAT                                                             70101510
      GO TO (4004,4005),JAC                                                70101520
4005  X=TPP(IL)                                                            70101530
      GO TO 103                                                            70101540
4004  Z=-AK*(TPP(IL)-TO)                                                   70101550
      X=EXP(Z)                                                             70101560
103   JP=4                                                                 70101570
      DO 15 K=1,4                                                          70101580
      IF(X-X4(K)) 16,16,15                                                 70101590
16    JP=K                                                                 70101600
      GO TO 25                                                             70101610
15    CONTINUE                                                             70101620
25    AREA1=0                                                              70101630
      ARA1=0                                                               70101640
      ARA2=0                                                               70101650
      ARA3=0                                                               70101660
      ARA4=0                                                               70101670
      ARA5=0                                                               70101680
      ARA6=0                                                               70101690
      ARA7=0                                                               70101700
      GO TO (29,30,31,32),JP                                               70101710
32    I=4                                                                  70101720
      ARA4=(X**(P+1.0))*(((X**3/(4.0+P))*F1234(I))-((X**2/(3.0+P))*FY12370101730
     1(I))+((X/(2.0+P))*FYY12(I))-((1.0/(1.0+P))*FXY12(I)))               70101740
      X=X4(3)                                                              70101750
      ARA5=(X**(P+1.0))*(((X**3/(4.0+P))*F1234(I))-((X**2/(3.0+P))*FY12370101760
     1(I))+((X/(2.0+P))*FYY12(I))-((1.0/(1.0+P))*FXY12(I)))               70101770
      ARA4=ARA4-ARA5                                                       70101780
31    I=3                                                                  70101790
      ARA3=(X**(P+1.0))*(((X**3/(4.0+P))*F1234(I))-((X**2/(3.0+P))*FY12370101800
     1(I))+((X/(2.0+P))*FYY12(I))-((1.0/(1.0+P))*FXY12(I)))               70101810
      X=X4(2)                                                              70101820
      ARA6=(X**(P+1.0))*(((X**3/(4.0+P))*F1234(I))-((X**2/(3.0+P))*FY12370101830
     1(I))+((X/(2.0+P))*FYY12(I))-((1.0/(1.0+P))*FXY12(I)))               70101840
      ARA3=ARA3-ARA6                                                       70101850
30    I=2                                                                  70101860
      ARA2=(X**(P+1.0))*(((X**3/(4.0+P))*F1234(I))-((X**2/(3.0+P))*FY12370101870
     1(I))+((X/(2.0+P))*FYY12(I))-((1.0/(1.0+P))*FXY12(I)))               70101880
      X=X4(1)                                                              70101890
      ARA7=(X**(P+1.0))*(((X**3/(4.0+P))*F1234(I))-((X**2/(3.0+P))*FY12370101900
     1(I))+((X/(2.0+P))*FYY12(I))-((1.0/(1.0+P))*FXY12(I)))               70101910
      ARA2=ARA2-ARA7                                                       70101920
```

```
29    I=1                                                                    70101930
      ARA1=(X**(P+1.0))*(((X**3/(4.0+P))*F1234(I))-((X**2/(3.0+P))*FY12370101940
     1(I))+((X/(2.0+P))*FYY12(I))-((1.0/(1.0+P))*FXY12(I)))                  70101950
      AREA(JK,IL)=ARA1+ARA2+ARA3+ARA4                                        70101960
      GO TO (80,99),KAT                                                      70101970
99    GO TO (80,77,78),KATY                                                  70101980
77    X=XINT                                                                 70101990
      TEMP=AREA(JK,IL)                                                       70102000
      KATY=3                                                                 70102010
      GO TO 103                                                              70102020
78    AREA(JK,IL)=TEMP-AREA(JK,IL)                                           70102030
      TEMP=0                                                                 70102040
80    CONTINUE                                                               70102050
93    CONTINUE                                                               70102060
      IST=IST+1                                                              70102070
      GO TO (127,128),IST                                                    70102080
127   DELTA=DDELTA                                                           70102090
      DO 222 I=1,N                                                           70102100
      DO 222 J=1,M                                                           70102110
222   SAVE(I,J)=AREA(I,J)                                                    70102120
      IF (DDELTA) 3001,3002,3001                                             70102130
3002  IILG=2                                                                 70102140
      GO TO 3004                                                             70102150
3001  IILG=1                                                                 70102160
      GO TO 203                                                              70102170
128   DO 223 I=1,N                                                           70102180
      DO 223 J=1,M                                                           70102190
      COMP1(I,J)=ABS(SAVE(I,J)-AREA(I,J))                                    70102200
      COMP2(I,J)=COMP1(I,J)*YW(I,J)                                          70102210
223   COMP1(I,J)=COMP1(I,J)/AREA(I,J)                                        70102220
3004  DO 224 I=1,N                                                           70102230
      DO 224 J=1,M                                                           70102240
      SAVE(I,J)=SAVE(I,J)*YW(I,J)                                            70102250
224   AREA(I,J)=AREA(I,J)*YW(I,J)                                            70102260
      IST=0                                                                  70102270
      WRITE (6,1011)D                                                        70102280
1011  FORMAT(1H ////34H OUTPUT BLOCK WITH INITIAL DELTA= F8.4)                70102290
      WRITE (6,1012)N,FO,RANGE                                               70102300
1012  FORMAT(88H THE T IN THE FIRST COLUMN REPRESENTS DISTANCE ALONG THE70102310
     1 T-AXIS (FOR THE BETA-INTEGRAL)./5H THE I2,94H NUMBERS OPPOSITE TH70102320
     2E T-COLUMN ARE THE EVALUATIONS OF THE FUNCTION AT THE LATTICE POIN70102330
     3TS (F,T)/19H WHERE F GOES FROM F8.4,4H TO F8.4)                        70102340
      DO 228 J=1,M                                                           70102350
228   WRITE (6,239)TPP(J),(SAVE(I,J),I=1,N)                                  70102360
239   FORMAT(1H /4H (T=F6.3,2H) ,2X,7(2XE12.5)/(14X,7E14.5))                 70102370
      GO TO (3005,3006),IILG                                                 70102380
3005  WRITE (6,1013)DDELTA                                                   70102390
1013  FORMAT(1H ////26H OUTPUT BLOCK WITH DELTA= F8.4)                        70102400
      WRITE (6,1012)N,FO,RANGE                                               70102410
      DO 229 J=1,M                                                           70102420
229   WRITE (6,239)TPP(J),(AREA(I,J),I=1,N)                                  70102430
      WRITE (6,1014)                                                         70102440
1014  FORMAT(1H ////74H OUTPUT BLOCK OF ABSOLUTE ERRORS BETWEEN THE FIRS70102450
     1T AND SECOND COMPUTATIONS)                                            70102460
      DO 230 J=1,M                                                           70102470
230   WRITE (6,239)TPP(J),(COMP2(I,J),I=1,N)                                 70102480
      WRITE (6,1015)                                                         70102490
1015  FORMAT(1H ////74H OUTPUT BLOCK OF RELATIVE ERRORS BETWEEN THE FIRS70102500
     1T AND SECOND COMPUTATIONS)                                            70102510
      DO 231 J=1,M                                                           70102520
231   WRITE (6,239)TPP(J),(COMP1(I,J),I=1,N)                                 70102530
3006  GO TO 606                                                              70102540
 9099 STOP
      END                                                                    70102550
```

Program name - FRG 708

PIECE-WISE INTEGRATION OF YIELD CURVES

Purpose

Performs piecewise integration of yield curves when the lifespan is divided into intervals and different mortality and growth rates are associated with each interval. An array for graphing a yield isopleth is produced.

Programmer - Lawrence E. Gales

Originating organization - Fisheries Research Institute, University of Washington

Approximate storage - Probably under 10,000

This program will compute an approximate yield isopleth for a given number of recruits to a fishery when both growth and natural mortality are estimated empirically. The calculations are carried out using a modified form of Ricker's method for estimating equilibrium yield (see Ricker, 1957, pp. 208-217). The program is extremely general in that growth, natural mortality and fishing mortality rates need not be measured using the same time intervals. Fishing mortality rates can be age specific (up to 400 different rates can be applied during the life of the fish) but the overall level of fishing mortality can be varied by means of multipliers which apply to all of the individual age specific rates. The range and the intervals between ages at first capture can also be varied by the user.

The program has two approximation options: (1) an exponential mode which assumes that the biomass of the stock changes in a strictly exponential manner during any interval when growth, natural mortality and fishing rates are all constant (see equation 10.4, Ricker, 1958); or (2) an arithmetic mode which uses the arithmetic mean of the stock biomass at the start and at the end of any interval during which all three rates are constant as an estimate of the average biomass present during the interval (see equation 10.3, Ricker, 1958).

Input to the program can be given in the form of instantaneous rates for growth, natural mortality and fishing mortality or the program can compute the instantaneous growth and natural mortality rates from weight-at-time input and from numbers-at-time input. If the latter form of input is employed the instantaneous average growth rate during the i^{th} period is calculated from the formula:

$$g_i = \ln (W_{t_{i+1}}/N_{t_i})/(t_{i+1} - t_i)$$

where t_i = time at which the weight of an individual fish is exactly W_{t_i} weight units

W_{t_i} = weight of an individual fish at time t_i

Instantaneous natural mortality rates are calculated similarly from the formula:

$$M_i = \ln (N_{t'_{i+1}}/N_{t'_i})/(t'_{i+1} - t'_i)$$

where t'_i = time at which the number of fish in the stock is exactly N_{t_i} pieces if natural mortality is the only source of mortality present

M_i = instantaneous natural mortality rate during the i^{th} time interval

$N_{t'_i}$ = number of fish in the stock at time t'_i when natural mortality is the only source of mortality

The program will compute and print out at specified times the biomass of the stock when only natural mortality and growth are present. This biomass vector is useful for determining the optimum harvest times for stocks (e.g. oyster stocks) that may be completely harvested at one time.

INPUT

The input consists of (I) a parameter or control card plus (II) a maximum of nine sets of data cards each of which, except for the last, is followed by a blank card.

I. Parameter Card

A. Purpose:

This card reads in four parameters (NG, NM, NOPT, NOPT1) which control input and output, plus an output label, and is punched according to FORMAT (4I1, 11A6).

B. Card format:

LOCATION ON CARD	FORTRAN SYMBOL	PURPOSE
Col. 1	NG	If 1 read in vector WT If 2 read in vector G
Col. 2	NM	If 1 read in vector ANT If 2 read in vector AM
Col. 3	NOPT	If 1 compute Y_A - yield arithmetic mode If 2 compute Y_E - yield exponential mode
Col. 4	NOPT1	If 1 skip sets 6 and 7, output biomass only If 2 output biomass and yield matrix If 3 output yield matrix only
Col. 5-70	COMNTS	Any desired label

II. Data Cards

A. Set descriptions and mathematical and FORTRAN equivalents:

SET	DESCRIPTION	MATH EQUIVALENT	FORTRAN EQUIVALENT
1	Divisions of growth scale	t	T
2	Weight at growth scale divisions (NG=1) or Instantaneous growth rates (NG = 2)	W_t g	WT G
3	Divisions of natural mortality scale	t'	TP
4	Numbers for natural mortality computations (NM=1) or Instantaneous natural mortality rates (NM=2)	$N_{t'}$ M	ANT AM
5	Divisions of fishing scale	t''	TPP
6	Instantaneous fishing mortality rates	F	F
7	Multipliers	m	AAM
8	Values of Tau (age at entry)	T	TAU
9	Number at start and Average weight at start	N_I W_I	ANI AWI

where all mathematical and FORTRAN symbols except for N_I, W_I, ANI, AWI stand for the vectors whose subscripted components are described in PURPOSE, e.g. t stands for the vector $(t_1, \ldots, t_n)$ where t_i = a division of growth scale = time at which weight of an individual fish is exactly W_{t_i} weight units, and T = (T(1), ..., T(n)) is its FORTRAN equivalent.

B. Card format and sequence:

1. Format:

(a) SETS 1-8

For SETS 1-8 the format is 9F8.0, i.e. 9 numbers per card each of which is a maximum of 7 digits long plus a decimal point.

The data for each set consists of a vector of numbers $V_1, \ldots, V_n$ punched according to this format. $V_1, \ldots, V_9$ are punched in columns 1-72 (V_1 in cols. 1-8, V_2 in cols. 9-16, ..., V_9 in cols. 65-72) on one card, $V_{10}, \ldots, V_{18}$ are punched in cols. 1-72 on the next, and so on until this vector is exhausted. A blank card is placed after this set and another vector for the next set is punched in the same manner as the above. For example, if in SET 1, n = 20, then $t_1, \ldots, t_9$ are punched in cols. 1-72 on one card, $t_{10}, \ldots, t_{18}$ in cols. 1-72 on the next, and t_{19}, t_{20} in cols. 1-8 and 9-16 on the third.

(b) SET 9

SET 9 consists of two numbers ANI and AWI punched according to FORMAT (2F10.0), i.e. ANI is punched in cols. 1-10, and AWI in cols. 11-20 of the same card.

2. Sequence:

If NOPT1 = 2 or 3 the sequence is as follows:

CARD NO.	DATA	FORTRAN VECTOR
CARD 1	PARAMETER CARD	NG, NM, NOPT, NOPT1, COMNTS
CARD 2 ... CARD J_1	DATA FOR SET 1	T
CARD J_1+1	BLANK CARD	
CARD J_1+2 ... CARD J_2	DATA FOR SET 2	WT or G
CARD J_2+1	BLANK CARD	
CARD J_2+2 ... CARD J_3	DATA FOR SET 3	TP
CARD J_3+1	BLANK CARD	
CARD J_3+2 ... CARD J_4	DATA FOR SET 4	ANT or AM
CARD J_4+1	BLANK CARD	

CARD J_4+2 ... CARD J_5	DATA FOR SET 5	TPP
CARD J_5+1	BLANK CARD	
CARD J_5+2 ... CARD J_6	DATA FOR SET 6	F
CARD J_6+1	BLANK CARD	
CARD J_6+2 ... CARD J_7	DATA FOR SET 7	AAM
CARD J_7+1	BLANK CARD	
CARD J_7+2 ... CARD J_8	DATA FOR SET 8	TAU
CARD J_8+1	BLANK CARD	
CARD J_8+2	DATA FOR SET 9	ANI, AWI

IF NOPT1 = 1 the data within the dotted lines, i.e. from CARD J_5+2 to CARD J_7+1, is deleted.

This completes the data for one problem. Additional problems may be handled simply by adding other groups of data cards in the same manner as above.

LIMITATIONS

The vectors T, WT, G, TP, ANT and AM can have a maximum of 99 components, TPP and F a maximum of 400 components, and AAM and TAU a maximum of 20 components.

INPUT AND OUTPUT FOR A SAMPLE PROBLEM

I. Input

(a) Numerical values:

$t_1 = 0$	$g_1 = .258$	$t'_1 = 0$	$M_1 = .025$	$t''_1 = 0$	$F_1 = 1$	$m_1 = .05$	$T_1 = 0$	$N_I = 1000$
$t_2 = 1$	$g_2 = .151$	$t'_2 = 1$	$M_2 = .025$	$t''_2 = 1$	$F_2 = 1$	$m_2 = .10$	$T_2 = 1$	$W_I = 1$
$t_3 = 2$	$g_3 = .095$	$t'_3 = 2$	$M_3 = .025$	$t''_3 = 2$	$F_3 = 1$	$m_3 = .15$	$T_3 = 2$	
$t_4 = 3$	$g_4 = .063$	$t'_4 = 3$	$M_4 = .025$	$t''_4 = 3$	$F_4 = 1$	$m_4 = .20$	$T_4 = 3$	
$t_5 = 4$		$t'_5 = 4$		$t''_5 = 4$		$m_5 = .25$	$T_5 = 4$	

where $(t_1, \ldots, t_5) = t \equiv T$

$(g_1, \ldots, g_4) = g \equiv G$

$(t'_1, \ldots, t'_5) = t' \equiv TP$

$(M_1, \ldots, M_4) = M \equiv AM$

$(t''_1, \ldots, t''_5) = t'' \equiv TPP$

$(F_1, \ldots, F_4) = F \equiv F$

$(m_1, \ldots, m_5) = m \equiv AAM$

$(T_1, \ldots, T_5) = T \equiv TAU$

$N_I \equiv ANI$

$W_I \equiv AWI$

and "$\equiv$" means "is the FORTRAN equivalent to".

(b) Options and label:

OPTION	PURPOSE
NG=2	Program reads in G
NM=2	Program reads in AM
NOPT=2	Program computes yield (Y_E) in exponential mode
NOPT1=2	Program outputs biomass and yield matrix

LABEL: The label will read "EXPONENTIAL MODE"

(c) Card form and sequence:

	NG	NM	NOPT	NOPT1	COMNTS
CARD 1	2	2	2	2	EXPONENTIAL MODE
cols.	1	2	3	4	5 — 20

	T(1)	T(2)	T(3)	T(4)	T(5)
CARD 2	0.	1.	2.	3.	4.
cols.	1 — 8	9 — 16	17—24	25—32	33—40

CARD 3 BLANK CARD

	G(1)	G(2)	G(3)	G(4)
CARD 4	.258	.151	.095	.063
cols.	1 — 8	9 — 16	17—24	25—32

CARD 5 BLANK CARD

FRG 708

	TP(1)	TP(2)	TP(3)	TP(4)	TP(5)
CARD 6	0.	1.	2.	3.	4.
cols.	1 — 8	9 — 16	17—24	25—32	33—40

CARD 7 BLANK CARD

	AM(1)	AM(2)	AM(3)	AM(4)
CARD 8	.025	.025	.025	.025
cols.	1 — 8	9 — 16	17—24	25—32

CARD 9 BLANK CARD

	TPP(1)	TPP(2)	TPP(3)	TPP(4)	TPP(5)
CARD 10	0.	1.	2.	3.	4.
cols.	1 — 8	9 — 16	17—24	25—32	33—40

CARD 11 BLANK CARD

	F(1)	F(2)	F(3)	F(4)
CARD 12	1.	1.	1.	1.
cols.	1 — 8	9 — 16	17—24	25—32

CARD 13 BLANK CARD

	AAM(1)	AAM(2)	AAM(3)	AAM(4)	AAM(5)
CARD 14	.05	.10	.15	.20	.25
cols.	1 — 8	9 — 16	17—24	25—32	33—40

CARD 15 BLANK CARD

	TAU(1)	TAU(2)	TAU(3)	TAU(4)	TAU(5)
CARD 16	0.	1.	2.	3.	4.
cols.	1 — 8	9 — 16	17—24	25—32	33—40

CARD 17 BLANK CARD

	ANI	AWI
CARD 18	1000.	1.
cols.	1 — 10	11—20

FRG 708

II. Output

```
**********************************************************************************

EXPONENTIAL MODE
PIECE-WISE INTEGRATION OF THE YIELD CURVE

                    - - - - - - - - - - - - - - - - - -
                          I N P U T   D A T A
                    - - - - - - - - - - - - - - - - - -

DIVISIONS OF GROWTH SCALE, 9 PER LINE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(N= 1)      0            1.0000       2.0000       3.0000       4.0000

INSTANTANEOUS GROWTH RATE, 9 PER LINE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(N= 1)      0.2580       0.1510       0.9500E-01   0.6300E-01   -0

DIVISIONS OF NATURAL MORTALITY SCALE, 9 PER LINE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(N= 1)      0            1.0000       2.0000       3.0000       4.0000

INSTANTANEOUS NATURAL MORTALITY RATES, 9 PER LINE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(N= 1)      0.2500E-01   0.2500E-01   0.2500E-01   0.2500E-01   -0

DIVISIONS OF FISHING SCALE, 9 PER LINE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(N= 1)      0            1.0000       2.0000       3.0000       4.0000

INSTANTANEOUS FISHING MORTALITY RATES, 9 PER LINE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(N= 1)      1.0000       1.0000       1.0000       1.0000       -0

MULTIPLIERS, 9 PER LINE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(N= 1)      0.5000E-01   0.1000       0.1500       0.2000       0.2500

VALUES OF TAU, 9 PER LINE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(N= 1)      0            1.0000       2.0000       3.0000       4.0000

NUMBER AT START = 1000.000                AVERAGE WEIGHT AT START = 1.000000

**********************************************************************************

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

BIOMASS VECTOR NO HARVEST

NUMBER	TIME	BIOMASS
1	0	1000.000
2	1.000000	1262.381
3	2.000000	1431.897
4	3.000000	1535.721
5	4.000000	1595.201

YIELD MATRIX

T(5)=	4.000	0	0	0	0	0
T(4)=	3.000	76.3271	148.908	217.926	283.556	345.963
T(3)=	2.000	144.920	275.801	393.989	500.702	597.042
T(2)=	1.000	203.432	377.448	526.212	653.301	761.790
T(1)=	0.	248.378	448.484	609.316	738.214	841.169

MULTIPLIER...

M(1)= 0.50000E-01
M(2)= 0.10000
M(3)= 0.15000
M(4)= 0.20000
M(5)= 0.25000

Additions to FRG 708 Write-up

The user is cautioned that all values taken on by TAU, the age at entry, must be represented in the set of TPP values or the isopleth computations will be in error. The program sets all of the instantaneous fishing mortality rates for ages less than a given age at entry equal to zero provided that a fishing mortality rate interval ends at the given age at entry.

For example, if the instantaneous fishing mortality rate is constant from age 2 to age 10 and TAU, the age of entry, is varied over the interval from age 2 to age 10 by one year steps; i.e. TAU = 2.0, 3.0, 4.0, ..., 10.0, the set of TPP's must also contain the values 2.0, 3.0, 4.0, ..., 10.0; if TPP only contains the values 2.0 and 10.0 all of the yield isopleth computations will be carried out exactly as if TAU were equal to 2.0 for all of the TAU's from 3.0 to 10.0. Since the program will accept 400 TPP's this restriction does not place any serious limitation on its use.

A more complete description of the generalized version of Ricker's equilibrium yield model and its use is given by Paulik and Bayliff (1967).

Special notes

This program reads from unit 5 and writes on unit 6. An end-of-file check is located at statement 1001+1 and the program should continue to operate properly after its removal.

A small assembly language function subroutine is utilized by this program and is listed in CDC 6400 COMPASS code at the end of the FORTRAN listing. This subroutine is used to search for blank and non-blank fields on the data cards.

It is in the logical mode and takes the value .TRUE. when the call is to BLANK and a blank field is encountered; when the field is non-blank it takes the value .FALSE.. Conversely when NONBL is called, .TRUE. is returned if a non-blank field is encountered and .FALSE. when a blank field is tested.

To use the program on other computer models this subroutine can be replaced with one written in the appropriate assembly language, the card reading sections of the FORTRAN program can be re-written to eliminate the need for checking blank fields, or on some computers the assembly language subroutine can be replaced with FORTRAN IV statement functions. The latter option is applicable if the computer stores blanks read under a numeric format as negative zeros and if the compiler's logical expressions can distinguish negative zeros from positive zeros. Since this change is easy to make and will frequently handle the problem it is suggested that it be tried first. The statement functions to be inserted directly after the dimension statement are:

```
BLANK(X) = X.LT.0.
NONBL(X) = X.GE.0.
```

References

Paulik, G.J. and W.F. Bayliff, A generalized computer program for the Ricker model of equilibrium
1967 yield per recruitment. J.Fish Res.Bd Canada, 24:249-252.

Ricker, W.E., Hand book of computations for biological statistics of fish populations. Fish.Res.
1958 Bd Canada Bull., (119):300 pp.

FRG 708

```
      PROGRAM F708(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C  FORTRAN IV          PIECE-WISE INTEGRATION OF YIELD CURVES
C
C                A GENERALIZED COMPUTER PROGRAM FOR THE RICKER MODEL
C                        OF EQUILIBRIUM YIELD PER RECRUITMENT
C
C
      LOGICAL BLANK,NONBL
      DIMENSION YE(20,20),YA(20,20),T(240),TP(240),WT(240),G(240),ANT(4470800040
     10),AM(440),TPP(540),FP(540),F(540),AAM(140),TAU(140),TTT(740),W(2470800050
     20)                                                                70800060
     3,COMNTS(11)                                                       70800070
     4,TPT(440),MROW(20)                                                70800080
C     DIMENSION YE(20,20),YA(20,20),T(240),TP(240),WT(240),G(240),ANT(4470800090
C    10),AM(440),TPP(540),FP(540),F(540),AAM(140),TAU(140),TTT(740),W(2470800100
C    20)                                                                70800110
C    3,COMNTS(11)                                                       70800120
C    4,TPT(440),MROW(20)                                                70800130
      DO 541 I=1,20                                                     70800140
  541 MROW(I)=I                                                         70800150
 1001 READ (5,8)NG,NM,NOPT,NOPT1,COMNTS                                 70800160
      IF(EOF,5) 998,999
  998 STOP
  999 CONTINUE
      DO 480 I=1,601                                                    70800170
  480 TTT(I)=0                                                          70800180
      IC=1                                                              70800190
      WRITE (6,230)COMNTS                                               70800200
      IE=9                                                              70800210
      WRITE (6,681)                                                     70800220
      WRITE (6,200)                                                     70800230
      WRITE (6,682)                                                     70800240
    4 READ (5,1)(T(I),I=IC,IE)                                          70800250
      IF(BLANK(T(IC))) GO TO 3
    2 WRITE (6,201)IC,(T(I),I=IC,IE)                                    70800270
      IC=IC+9                                                           70800280
      IE=IE+9                                                           70800290
      GO TO 4                                                           70800300
    3 ICC=IC-9                                                          70800310
      ICT=IC                                                            70800320
      BBB=T(ICT)                                                        70800330
      DO 5I=ICC,IC                                                      70800340
      IF(NONBL(T(I)))GO TO 5
    6 MAXT=I-1                                                          70800360
      GO TO 7                                                           70800370
    5 CONTINUE                                                          70800380
    7 IC=1                                                              70800390
      WRITE (6,681)                                                     70800400
      IE=9                                                              70800410
      GO TO (231,232),NG                                                70800420
  231 WRITE (6,102)                                                     70800430
      WRITE (6,682)                                                     70800440
      GO TO 14                                                          70800450
  232 WRITE (6,233)                                                     70800460
      WRITE (6,682)                                                     70800470
   14 READ (5,11)(G(I),I=IC,IE)                                         70800480
      IF(BLANK(G(IC)))GO TO 13
 12   WRITE (6,203)IC,(G(I),I=IC,IE)                                    70800500
      IC=IC+9                                                           70800510
      IE=IE+9                                                           70800520
      GO TO 14                                                          70800530
```

PRG 708

```
   I3 ICC=IC-9                                                          70800540
      ICCC=IC                                                           70800550
      DO 15 I=ICC,IC                                                    70800560
      IF(NONBL(G(I)))GO TO 15
   16 MAX G=I-1                                                         70800580
      GO TO 17                                                          70800590
   15 CONTINUE                                                          70800600
   17 GO TO (9,10),NG                                                   70800610
    9 DO 18 I=1,MAXG                                                    70800620
   18 WT(I)=G(I)                                                        70800630
      MAXG=MAXG-1                                                       70800640
      DO 19I=1,MAXG                                                     70800650
   19 G(I)=(ALOG(WT(I+1)/WT(I)))/(T(I+1)-T(I))                          70800660
      DO 2001 I=1,28                                                    70800670
      JL=MAXG+I                                                         70800680
2001  G(JL)=BBB                                                         70800690
      WRITE (6,681)                                                     70800700
      WRITE (6,233)                                                     70800710
      WRITE (6,682)                                                     70800720
      IC=1                                                              70800730
      IE=9                                                              70800740
   24 IF(BLANK(G(IC)))GO TO 23
   22 WRITE (6,203)IC,(G(I),I=IC,IE)                                    70800760
      IC=IC+9                                                           70800770
      IE=IE+9                                                           70800780
      GO TO 24                                                          70800790
   23 CONTINUE                                                          70800800
   10 IC=1                                                              70800810
      IE=9                                                              70800820
      WRITE (6,681)                                                     70800830
      WRITE (6,205)                                                     70800840
      WRITE (6,682)                                                     70800850
   34 READ (5,1)(TP(I),I=IC,IE)                                         70800860
      IF(BLANK(TP(IC)))GO TO 33
   32 WRITE (6,201)IC,(TP(I),I=IC,IE)                                   70800880
      IC=IC+9                                                           70800890
      IE=IE+9                                                           70800900
      GO TO 34                                                          70800910
   33 ICC=IC-9                                                          70800920
      ICTP=IC                                                           70800930
      DO 35 I=ICC,IC                                                    70800940
      IF(NONBL(TP(I)))GO TO 35
   36 MAXTP=I-1                                                         70800960
      GO TO 37                                                          70800970
   35 CONTINUE                                                          70800980
   37 IC=1                                                              70800990
      IE=9                                                              70801000
      WRITE (6,681)                                                     70801010
      GO TO (235,236),NM                                                70801020
235   WRITE (6,207)                                                     70801030
      WRITE (6,682)                                                     70801040
      GO TO 44                                                          70801050
  236 WRITE (6,237)                                                     70801060
      WRITE (6,682)                                                     70801070
44    READ (5,11)(AM(I),I=IC,IE)                                        70801080
      IF(BLANK(AM(IC)))GO TO 43
   42 WRITE (6,203)IC,(AM(I),I=IC,IE)                                   70801100
      IC=IC+9                                                           70801110
      IE=IE+9                                                           70801120
      GO TO 44                                                          70801130
   43 ICC=IC-9                                                          70801140
      ICCC=IC                                                           70801150
      DO 45 I=ICC,IC                                                    70801160
      IF(NONBL(AM(I)))GO TO 45
```

```
   46 MAXM=I-1                                                          70801180
      GO TO 47                                                          70801190
   45 CONTINUE                                                          70801200
   47 GO TO (419,50),NM                                                 70801210
  419 DO 48 I=1,MAXM                                                    70801220
   48 ANT(I)=AM(I)                                                      70801230
      MAXM=MAXM-1                                                       70801240
      DO 49 I=1,MAXM                                                    70801250
   49 AM(I)=-ALOG((ANT(I+1)/ANT(I)))/(TP(I+1)-TP(I))                    70801260
      DO 2003 I=1,28                                                    70801270
      JL=MAXM+I                                                         70801280
2003  AM(JL)=BBB                                                        70801290
      WRITE (6,681)                                                     70801300
      WRITE (6,237)                                                     70801310
      WRITE (6,682)                                                     70801320
      IC=1                                                              70801330
      IE=9                                                              70801340
   54 IF(BLANK(AM(IC)))GO TO 53                                         
   52 WRITE (6,203)IC,(AM(I),I=IC,IE)                                   70801360
      IC=IC+9                                                           70801370
      IE=IE+9                                                           70801380
      GO TO 54                                                          70801390
53    CONTINUE                                                          70801400
   50 IC=1                                                              70801410
      IE=9                                                              70801420
      WRITE (6,681)                                                     70801430
      WRITE (6,212)                                                     70801440
      WRITE (6,682)                                                     70801450
   64 READ (5,1)(TPP(I),I=IC,IE)                                        70801460
      IF(BLANK(TPP(IC)))GO TO 63                                        
   62 WRITE (6,201)IC,(TPP(I),I=IC,IE)                                  70801480
      IC=IC+9                                                           70801490
      IE=IE+9                                                           70801500
      GO TO 64                                                          70801510
   63 ICC=IC-9                                                          70801520
      DO 65 I=ICC,IC                                                    70801530
      IF(NONBL(TPP(I)))GO TO 65                                         
   66 MAXTPP=I-1                                                        70801550
      GO TO 67                                                          70801560
   65 CONTINUE                                                          70801570
   67 IC=1                                                              70801580
      GO TO (87,260,260),NOPT1                                          70801590
  260 IE=9                                                              70801600
      WRITE (6,681)                                                     70801610
      WRITE (6,214)                                                     70801620
      WRITE (6,682)                                                     70801630
   74 READ (5,11)(FP(I),I=IC,IE)                                        70801640
      IF(BLANK(FP(IC)))GO TO 73                                         
   72 WRITE (6,203)IC,(FP(I),I=IC,IE)                                   70801660
      IC=IC+9                                                           70801670
      IE=IE+9                                                           70801680
      GO TO 74                                                          70801690
   73 ICC=IC-9                                                          70801700
      DO 75 I=ICC,IC                                                    70801710
      IF(NONBL(FP(I)))GO TO 75                                          
   76 MAXFP=I-1                                                         70801730
      GO TO 77                                                          70801740
   75 CONTINUE                                                          70801750
   77 IC=1                                                              70801760
      IE=9                                                              70801770
      WRITE (6,681)                                                     70801780
      WRITE (6,216)                                                     70801790
      WRITE (6,682)                                                     70801800
```

```
   84 READ (5,81)(AAM(I),I=IC,IE)                                    70801810
      IF(BLANK(AAM(IC)))GO TO 83
   82 WRITE (6,217)IC,(AAM(I),I=IC,IE)                               70801830
      IC=IC+9                                                        70801840
      IE=IE+9                                                        70801850
      GO TO 84                                                       70801860
   83 ICC=IC-9                                                       70801870
      DO 85 I=ICC,IC                                                 70801880
      IF(NONBL(AAM(I)))GO TO 85
   86 MAXAAM=I-1                                                     70801900
      GO TO 87                                                       70801910
   85 CONTINUE                                                       70801920
   87 IC=1                                                           70801930
      IE=9                                                           70801940
      WRITE (6,681)                                                  70801950
      WRITE (6,220)                                                  70801960
      WRITE (6,682)                                                  70801970
   94 READ (5,1)(TAU(I),I=IC,IE)                                     70801980
      IF(BLANK(TAU(IC)))GO TO 93
   92 WRITE (6,201)IC,(TAU(I),I=IC,IE)                               70802000
      IC=IC+9                                                        70802010
      IE=IE+9                                                        70802020
      GO TO 94                                                       70802030
   93 ICC=IC-9                                                       70802040
      DO 95 I=ICC,IC                                                 70802050
      IF(NONBL(TAU(I)))GO TO 95
   96 MAXTAU=I-1                                                     70802070
      GO TO 97                                                       70802080
   95 CONTINUE                                                       70802090
   97 READ (5,225)ANI,AWI                                            70802100
      WRITE (6,681)                                                  70802110
      WRITE (6,240)ANI,AWI                                           70802120
      ANW=ANI*AWI                                                    70802130
      MAXT1=MAXT+1                                                   70802140
      MAXTP1=MAXTP+1                                                 70802150
      MAXPP1=MAXTPP+1                                                70802160
      DO 2013 MIQ=1,5                                                70802170
      INDX1=MAXT+MIQ                                                 70802180
      INDX2=MAXTP+MIQ                                                70802190
      INDX3=MAXTPP+MIQ                                               70802200
      INDX4=MAXFP+MIQ                                                70802210
      T(INDX1)=0                                                     70802220
      TP(INDX2)=0                                                    70802230
      TPP(INDX3)=0                                                   70802240
2013  FP(INDX4)=0                                                    70802250
  350 CONTINUE                                                       70802260
 1000 DO 300 ITAU=1,MAXTAU                                           70802270
      DO 301 IM=1,MAXAAM                                             70802280
      GO TO (310,310,311),NOPT1                                      70802290
  310 DO 312 I=1,MAXPP1                                              70802300
  312 F(I)=0                                                         70802310
      GO TO 313                                                      70802320
311   DO 303 I=1,MAXTPP                                              70802330
      IF(TPP(I)-TAU(ITAU))303,304,305                                70802340
  304 K=I                                                            70802350
      GO TO 307                                                      70802360
  305 K=I-1                                                          70802370
      GO TO 307                                                      70802380
  303 CONTINUE                                                       70802390
      K=MAXTPP                                                       70802400
  307 K1=K-1                                                         70802410
      DO 308 I=1,K1                                                  70802420
```

```
  308 F(I)=0                                                              70802430
      DO 309 I=K,MAXTPP                                                   70802440
  309 F(I)=FP(I)*AAM(IM)                                                  70802450
  313 YE(ITAU,IM)=0                                                       70802460
      YA(ITAU,IM)=0                                                       70802470
      W(1)=ANW                                                            70802480
      K=1                                                                 70802490
      TTT(1)=T(1)                                                         70802500
      TT=T(1)                                                             70802510
      I=2                                                                 70802520
      IP=2                                                                70802530
      IPP=2                                                               70802540
   99 IF(I-MAXT1)112,116,116                                              70802550
  112 IF(IP-MAXTP1)113,116,116                                            70802560
  113 IF (IPP-MAXPP1)114,116,116                                          70802570
114   IF(T(I)-TP(IP)) 101,902,103                                         70802580
  101 IF(T(I)-TPP(IPP))104,105,106                                        70802590
902   IF(TP(IP)-TPP(IPP))107,108,106                                      70802600
  103 IF(TP(IP)-TPP(IPP))110,109,106                                      70802610
  106 K=K+1                                                               70802620
      DELTA=TPP(IPP)-TT                                                   70802630
      EX=G(I-1)-AM(IP-1)-F(IPP-1)                                         70802640
      W(K)=W(K-1)*(EXP(EX*DELTA))                                         70802650
      GO TO (400,401),NOPT                                                70802660
  400 YA(ITAU,IM)=YA(ITAU,IM)+(DELTA*F(IPP-1)*(W(K-1)+W(K)))              70802670
      GO TO 402                                                           70802680
  401 ABEX=ABS(EX)                                                        70802690
      IF(ABEX-.000001) 720,720,721                                        70802700
  720 Y=DELTA                                                             70802710
      GO TO 722                                                           70802720
  721 Y=((EXP(EX*DELTA))-1.0)/EX                                          70802730
  722 CONTINUE                                                            70802740
      YE(ITAU,IM)=YE(ITAU,IM)+(F(IPP-1)*W(K-1)*Y)                         70802750
  402 TT=TPP(IPP)                                                         70802760
      TTT(K)=TT                                                           70802770
      IPP=IPP+1                                                           70802780
      GO TO 99                                                            70802790
  104 K=K+1                                                               70802800
      DELTA=T(I)-TT                                                       70802810
      EX=G(I-1)-AM(IP-1)-F(IPP-1)                                         70802820
      W(K)=W(K-1)*(EXP(EX*DELTA))                                         70802830
      GO TO (410,411),NOPT                                                70802840
  410 YA(ITAU,IM)=YA(ITAU,IM)+(DELTA*F(IPP-1)*(W(K)+W(K-1)))              70802850
      GO TO 412                                                           70802860
  411 ABEX=ABS(EX)                                                        70802870
      IF(ABEX-.000001) 730,730,731                                        70802880
  730 Y=DELTA                                                             70802890
      GO TO 732                                                           70802900
  731 Y=((EXP(EX*DELTA))-1.0)/EX                                          70802910
  732 CONTINUE                                                            70802920
      YE(ITAU,IM)=YE(ITAU,IM)+(F(IPP-1)*W(K-1)*Y)                         70802930
  412 TT=T(I)                                                             70802940
      TTT(K)=TT                                                           70802950
      I=I+1                                                               70802960
      GO TO 99                                                            70802970
  105 K=K+1                                                               70802980
      DELTA=T(I)-TT                                                       70802990
      EX=G(I-1)-AM(IP-1)-F(IPP-1)                                         70803000
      W(K)=W(K-1)*(EXP(EX*DELTA))                                         70803010
      GO TO (420,421),NOPT                                                70803020
  420 YA(ITAU,IM)=YA(ITAU,IM)+(DELTA*F(IPP-1)*(W(K-1)+W(K)))              70803030
      GO TO 422                                                           70803040
```

PRG 708

```
421 ABEX=ABS(EX)                                                    70803050
    IF(ABEX-.000001) 740,740,741                                    70803060
740 Y=DELTA                                                         70803070
    GO TO 742                                                       70803080
741 Y=((EXP(EX*DELTA))-1.0)/EX                                      70803090
742 CONTINUE                                                        70803100
    YE(ITAU,IM)=YE(ITAU,IM)+(F(IPP-1)*W(K-1)*Y)                     70803110
422 TT=T(I)                                                         70803120
    TTT(K)=TT                                                       70803130
    I=I+1                                                           70803140
    IPP=IPP+1                                                       70803150
    GO TO 99                                                        70803160
107 K=K+1                                                           70803170
    DELTA=T(I)-TT                                                   70803180
    EX=G(I-1)-AM(IP-1)-F(IPP-1)                                     70803190
    W(K)=W(K-1)*(EXP(EX*DELTA))                                     70803200
    GO TO (440,441),NOPT                                            70803210
440 YA(ITAU,IM)=YA(ITAU,IM)+(DELTA*F(IPP-1)*(W(K-1)*W(K)))          70803220
    GO TO 442                                                       70803230
441 ABEX=ABS(EX)                                                    70803240
    IF(ABEX-.000001) 750,750,751                                    70803250
750 Y=DELTA                                                         70803260
    GO TO 752                                                       70803270
751 Y=((EXP(EX*DELTA))-1.0)/EX                                      70803280
752 CONTINUE                                                        70803290
    YE(ITAU,IM)=YE(ITAU,IM)+(F(IPP-1)*W(K-1)*Y)                     70803300
442 TT=T(I)                                                         70803310
    TTT(K)=TT                                                       70803320
    I=I+1                                                           70803330
    IP=IP+1                                                         70803340
    GO TO 99                                                        70803350
108 K=K+1                                                           70803360
    DELTA=T(I)-TT                                                   70803370
    EX=G(I-1)-AM(IP-1)-F(IPP-1)                                     70803380
    W(K)=W(K-1)*(EXP(EX*DELTA))                                     70803390
    GO TO (450,451),NOPT                                            70803400
450 YA(ITAU,IM)=YA(ITAU,IM)+(DELTA*F(IPP-1)*(W(K)+W(K-1)))          70803410
    GO TO 452                                                       70803420
451 ABEX=ABS(EX)                                                    70803430
    IF(ABEX-.000001) 760,760,761                                    70803440
760 Y=DELTA                                                         70803450
    GO TO 762                                                       70803460
761 Y=((EXP(EX*DELTA))-1.0)/EX                                      70803470
762 CONTINUE                                                        70803480
    YE(ITAU,IM)=YE(ITAU,IM)+(F(IPP-1)*W(K-1)*Y)                     70803490
452 TT=T(I)                                                         70803500
    TTT(K)=TT                                                       70803510
    IP=IP+1                                                         70803520
    IPP=IPP+1                                                       70803530
    I=I+1                                                           70803540
    GO TO 99                                                        70803550
109 K=K+1                                                           70803560
    DELTA=TP(IP)-TT                                                 70803570
    EX=G(I-1)-AM(IP-1)-F(IPP-1)                                     70803580
    W(K)=W(K-1)*(EXP(EX*DELTA))                                     70803590
    GO TO (460,461),NOPT                                            70803600
460 YA(ITAU,IM)=YA(ITAU,IM)+(DELTA*F(IPP-1)*(W(K)+W(K-1)))          70803610
    GO TO 462                                                       70803620
461 ABEX=ABS(EX)                                                    70803630
    IF(ABEX-.000001) 770,770,771                                    70803640
770 Y=DELTA                                                         70803650
    GO TO 772                                                       70803660
771 Y=((EXP(EX*DELTA))-1.0)/EX                                      70803670
```

```
  772 CONTINUE                                                          70803680
      YE(ITAU,IM)=YE(ITAU,IM)+(F(IPP-1)*W(K-1)*Y)                       70803690
  462 TT=TP(IP)                                                         70803700
      TTT(K)=TT                                                         70803710
      IP=IP+1                                                           70803720
      IPP=IPP+1                                                         70803730
      GO TO 99                                                          70803740
  110 K=K+1                                                             70803750
      DELTA=TP(IP)-TT                                                   70803760
      EX=G(I-1)-AM(IP-1)-F(IPP-1)                                       70803770
      W(K)=W(K-1)*(EXP(EX*DELTA))                                       70803780
      GO TO (470,471),NOPT                                              70803790
  470 YA(ITAU,IM)=YA(ITAU,IM)+(DELTA*F(IPP-1)*(W(K)+W(K-1)))            70803800
      GO TO 472                                                         70803810
  471 ABEX=ABS(EX)                                                      70803820
      IF(ABEX-.000001) 780,780,781                                      70803830
  780 Y=DELTA                                                           70803840
      GO TO 782                                                         70803850
  781 Y=((EXP(EX*DELTA))-1.0)/EX                                        70803860
  782 CONTINUE                                                          70803870
      YE(ITAU,IM)=YE(ITAU,IM)+(F(IPP-1)*W(K-1)*Y)                       70803880
  472 TT=TP(IP)                                                         70803890
      TTT(K)=TT                                                         70803900
      IP=IP+1                                                           70803910
      GO TO 99                                                          70803920
  116 GO TO (320,320,321),NOPT1                                         70803930
  320 WRITE (6,339)                                                     70803940
      DO 330 I=1,K                                                      70803950
  330 WRITE (6,340)I,TTT(I),W(I)                                        70803960
      WRITE (6,341)                                                     70803970
      GO TO (1001,342,342),NOPT1                                        70803980
  342 NOPT1=3                                                           70803990
      GO TO 350                                                         70804000
  321 CONTINUE                                                          70804010
  301 CONTINUE                                                          70804020
  300 CONTINUE                                                          70804030
      WRITE (6,498)                                                     70804040
      DO 520 J=1,MAXTAU                                                 70804050
      DO 520 K=1,MAXAAM                                                 70804060
      YA(J,K)=YA(J,K)/2.0                                               70804070
520   CONTINUE                                                          70804080
      GO TO (501,502),NOPT                                              70804090
  501 MMM=MAXTAU+1                                                      70804100
      K=1                                                               70804110
      DO 500 I=1,MAXTAU                                                 70804120
      J=MMM-I                                                           70804130
      IF (MAXAAM-7) 1220,1220,1221                                      70804140
1220  WRITE (6,1250)J,TAU(J),(YA(J,K),K=1,MAXAAM)                       70804150
      GO TO 500                                                         70804160
1221  WRITE (6,1250)J,TAU(J),(YA(J,K),K=1,7)                            70804170
      WRITE (6,1251)(YA(J,K),K=8,MAXAAM)                                70804180
500   CONTINUE                                                          70804190
      WRITE (6,670)                                                     70804200
      IF(MAXAAM-7)1240,1240,1241                                        70804210
1240  WRITE (6,1260)(MROW(I),I=1,MAXAAM)                                70804220
      GO TO 1245                                                        70804230
1241  WRITE (6,1260)(MROW(I),I=1,7)                                     70804240
      WRITE (6,1261)(MROW(I),I=8,MAXAAM)                                70804250
1245  CONTINUE                                                          70804260
      WRITE (6,3002)AAM(1)                                              70804270
      IF (MAXAAM-1) 3010,3010,3011                                      70804280
3011  CONTINUE                                                          70804290
      DO 3004 I=2,MAXAAM                                                70804300
      WRITE (6,3003)I,AAM(I)                                            70804310
```

```
3004  CONTINUE                                                          70804320
3010  CONTINUE                                                          70804330
      GO TO 1001                                                        70804340
  502 MMM=MAXTAU+1                                                      70804350
      K=1                                                               70804360
      DO 504 I=1,MAXTAU                                                 70804370
      J=MMM-I                                                           70804380
      IF (MAXAAM-7) 1230,1230,1231                                      70804390
1230  WRITE (6,1250)J,TAU(J),(YE(J,K),K=1,MAXAAM)                       70804400
      GO TO 504                                                         70804410
1231  WRITE (6,1250)J,TAU(J),(YE(J,K),K=1,7)                            70804420
      WRITE (6,1251)(YE(J,K),K=8,MAXAAM)                                70804430
504   CONTINUE                                                          70804440
      WRITE (6,670)                                                     70804450
      IF(MAXAAM-7)1270,1270,1271                                        70804460
1270  WRITE (6,1260)(MROW(I),I=1,MAXAAM)                                70804470
      GO TO 1275                                                        70804480
1271  WRITE (6,1260)(MROW(I),I=1,7)                                     70804490
      WRITE (6,1261)(MROW(I),I=8,MAXAAM)                                70804500
1275  CONTINUE                                                          70804510
      WRITE (6,3002)AAM(1)                                              70804520
      IF (MAXAAM-1) 3020,3020,3021                                      70804530
3021  CONTINUE                                                          70804540
      DO 3006 I=2,MAXAAM                                                70804550
3006  WRITE (6,3003)I,AAM(I)                                            70804560
3020  CONTINUE                                                          70804570
      GO TO 1001                                                        70804580
1260  FORMAT(16H MULTIPLIER ... ,4X,7(2HM(,I2,1H),8X))                  70804590
1261  FORMAT(20X,2HM(,I2,1H),8X,2HM(,I2,1H),8X,2HM(,I2,1H),8X,2HM(,I2,1H70804600
     1),8X,2HM(,I2,1H),8X,2HM(,I2,1H),8X,2HM(,I2,1H),8X)                70804610
1250  FORMAT(16X,1H/,/16X,1H/,/3H T(,I2,2H)=,F8.3,2H /,7(1X,E12.5))     70804620
1251  FORMAT(16X,1H/,7E13.5)                                            70804630
    8 FORMAT(4I1,11A6)                                                  70804640
    1 FORMAT(9F8.0)                                                     70804650
681   FORMAT(1H ///)                                                    70804660
 498  FORMAT(1H1,55(2H**)//40X,32(1H-)/44X,24H Y I E L D  M A T R I X /470804670
     10X,32(1H-)///)                                                    70804680
682   FORMAT(1H ,60(1H-))                                               70804690
  207 FORMAT(1H0,54HNUMBERS FOR NATURAL MORTALITY COMPUTATIONS, 9 PER LI70804700
     1NE/)                                                              70804710
  230 FORMAT(1H1,55(2H**)/1H ,11A6/1H ,41HPIECE-WISE INTEGRATION OF THE 70804720
     1YIELD CURVE//35X,28(1H-)/39X,20H I N P U T  D A T A /35X,28(1H-)) 70804730
201   FORMAT(4H (N=,I3,2H) ,9(1X,E11.4))                                70804740
233   FORMAT(1H0,38HINSTANTANEOUS GROWTH RATES, 9 PER LINE/)            70804750
200   FORMAT(1H0,26HDIVISIONS OF GROWTH SCALE,12H  9 PER LINE/)         70804760
203   FORMAT(4H (N=,I3,2H) ,9(1X,E11.4))                                70804770
  102 FORMAT(1H0,45HWEIGHTS AT GROWTH SCALE DIVISIONS, 9 PER LINE/)     70804780
   11 FORMAT(9F8.0)                                                     70804790
 3003 FORMAT(16X,2HM(,I2,2H)=,E13.5)                                    70804800
 3002 FORMAT(1H0,9X,12HWHERE M( 1)=,E13.5)                              70804810
  205 FORMAT(1H0,49HDIVISIONS OF NATURAL MORTALITY SCALE,  9 PER LINE/) 70804820
  237 FORMAT(1H0,49HINSTANTANEOUS NATURAL MORTALITY RATES, 9 PER LINE/) 70804830
  240 FORMAT(1H0,17HNUMBER AT START= E14.6,3X,25HAVERAGE WEIGHT AT START70804840
     1= E14.6/1H ,50(2H**))                                             70804850
  212 FORMAT(1H0,39HDIVISIONS OF FISHING SCALE,  9 PER LINE/)           70804860
  214 FORMAT(1H0,49HINSTANTANEOUS FISHING MORTALITY RATES, 9 PER LINE/) 70804870
  216 FORMAT(1H0,24HMULTIPLIERS,  9 PER LINE/)                          70804880
   81 FORMAT(9F8.0)                                                     70804890
217   FORMAT(4H (N=,I3,2H) ,9(1X,E11.4))                                70804900
 220  FORMAT(1H0,26HVALUES OF TAU,  9 PER LINE/)                        70804910
  225 FORMAT(2F10.0)                                                    70804920
  339 FORMAT(1H ////1H ,50(2H--)/1H 26HBIO-MASS VECTOR NO HARVEST//7H NU70804930
     1MBER,4X,4HTIME,11X,8HBIO-MASS/)                                   70804940
```

```
 340 FORMAT(3X,I4,2X,E13.6,3X,E13.6)                                          70804950
 341  FORMAT(1H0,55(2H--))                                                    70804960
 670  FORMAT(16X,94(1H-))                                                     70804970
  660 FORMAT(3H K=,I4,3H I=,I4,4H FP=,I4,5H,IPP=,I4,4H TT=,F9.4,4H  F=,F70804980
     19.4,4H  G=,F9.4,4H AM=,F9.4)                                           70804990
661   FORMAT(1H ,2HW(,I2,4H)=W(,I2,9H)*EXPF(G(,I2,4H)-M(,I2,4H)-F(,I2,3H70805000
     1))(,F8.4,1H),3X,2HG=F8.3,2X,2HM=F8.3,2X,2HF=F8.3,4X,2HW=F9.4)       70805010
      END                                                                    70805020
          IDENT  BLANDK
*         CDC VERSION    10/9/69
*         BLANK AND NONBL ARE LOGIGAL FUNCTIONS
*         TO DISTINGUISH BLANKS IN I OR F FORMAT
*
          ENTRY  BLANK
          ENTRY  NONBL
          VFD    42/0HBLANK,18/1
BLANK     BSS    1
          SA1    B1
          PL     X1,FALSE
          NZ     X1,FALSE
TRUE      MX6    60
          JP     BLANK
          VFD    42/0HNONBL,18/1
NONBL     BSS    1
          SA1    B1
          SA2    NONBL
          BX6    X2
          SA6    BLANK
          PL     X1,TRUE
          NZ     X1,TRUE
FALSE     SX6    B0
          JP     BLANK
          END
```

Program name - YIELD 2

Purpose

Performs piecewise integration of yield curves when age is unknown. Different mortality rates may be associated with intervals in the lifespan and growth is calculated as a function of length from a transformed von Bertalanffy growth curve. Yield isopleths are given as functions of length-at-entry and F. Note that program BGC 4 provides von Bertalanffy growth parameters from unaged fish which can be used with this program.

Programmer - Patrick K. Tomlinson

Originating organization - California Department of Fish and Game

Approximate storage - 15,000 (CDC 1604)

The amount of growth a fish will put on during an interval of time is a function of the size at the beginning of the interval, not the age. Similarly, survival is usually given as a function of time elapsed, not age. Therefore, growth put on during an interval and survival during the interval can be combined to produce yield, even though age is unknown.

Given a group of fish of equal starting length, additional growth in length can be divided into increments of arbitrary size. Although these increments may vary in size, the unit of measurement should remain the same.

Left:
- m = number of increments
- i = 1,2,...,m
- T_i = time required to grow from S_i to S_{i+1}
- S_i = fish length at the beginning of i
- N_i = number of fish at the beginning of i
- F_i = instantaneous fishing mortality during T_i
- M_i = instantaneous natural mortality during T_i
- C_i = number caught during T_i
- Y_i = weight caught during T_i
- W_i = Y_i/C_i = mean weight for T_i
- ΣC_i = total number caught
- ΣY_i = total weight caught
- $\Sigma Y_i/\Sigma C_i$ = mean weight
- $\Sigma Y_i/N_1$ = yield-per-recruit

If the fish are growing according to a von Bertalanffy-type curve, then data from tagged fish can be used to fit the following:

- l_t = $L_\infty + (l-L_\infty)\exp(-Kt)$; where
- l = length at tagging
- l_t = length at recovery
- t = time lapse from l to l_t

Once L_∞ and K are known, then $T_i = -\log[(S_{i+1}-L_\infty)/(S_i-L_\infty)]/K$ .... (1)
estimates time required to get from S_i to S_{i+1}.

YIELD 2

If the interval, T_i, is sufficiently short and a weight-length relation ($w = al^b$) exists then the mean weight, W_i, for fish caught during T_i is approximately:

$$W_i = a[(S_{i+1})^b + (S_i)^b]/2 \qquad \ldots\ldots (2)$$

Therefore, if L_∞, K, a, b, N_1 and the set $\{F_i, M_i, S_i,\ i = 1,2,\ldots,m\}$ are given, then the set $\{T_i, C_i, W_i, Y_i, N_i,\ i = 1,2,\ldots,m\}$ can be computed from equations (1) and (2); and

$$N_{i+1} = N_i \exp[-T_i(F_i+M_i)]$$

$$C_i = N_i F_i\ (1-\exp[-T_i(F_i+M_i)]/(F_i+M_i)$$

$$Y_i = C_i W_i$$

The computer program allows for changing length at entry by setting all $F_i = 0$ for $S_i < S_j$, where S_j is the entry length. The program allows for trial changes in fishing intensity by multiples.

The program produces four tables. Table 1 gives the set $\{i, S_i, N_i, F_i, M_i, T_i, C_i, W_i, Y_i,\ i = 1,2,\ldots,m\}$ and totals $\Sigma T_i F_i, \Sigma T_i M_i, \Sigma T_i, \Sigma C_i, \Sigma Y_i / \Sigma C_i$, and ΣY_i.

Table 2 gives values of ΣY_i for varying sets of $\{F_i\}$ when length at entry changes and various multipliers of F are tried.

Table 3 is like table 2, except ΣC_i is printed.

Table 4 gives the lengths $(S_i + S_{i+1})/2$, the number alive at these lengths, and the weight of the numbers. These 3 sets occur under conditions of no fishing.

The cards are defined in terms of sets.

Set #1	1 card	FORMAT (10A8)
	Col. 1	Numeral 1
	Cols. 2–80	Any title information desired. This will be printed in upper left corner of each table.
Set #2	1 card	FORMAT (5F10.0,I4)
	Cols. 1–10	Value of N_1 = initial number of fish. Punch decimal point.
	Cols. 11–20	Value of L_∞ = asymptotic length. Punch decimal point.
	Cols. 21–30	Value of K = coefficient of growth, from von Bertalanffy equation. Punch decimal point.
	Cols. 31–40	Value of a from $w = al^b$ curve. Punch decimal point.
	Cols. 41–50	Value of b from $w = al^b$ curve. Punch decimal point.
	Cols. 51–54	Value of m = no. of increments. Punch right-justified with no decimal point.
Set #3	$1 \leq$ cards ≤ 111	FORMAT (9F8.0)
	Cols. 1–72	Each card has 9 fields of 8 columns each. Values of $\{F_i\}$ are punched in sequence, with the first nine in card one, etc. until all F_i are used. Punch decimal point.
Set #4	$1 \leq$ cards ≤ 111	FORMAT (9F8.0)
	Cols. 1–72	This set like set 3, except each field contains a value of M_i. Punch decimal point.

Set #5	1 ≤ cards ≤ 112 Cols. 1-72	FORMAT (9F8.0) This set like set 3, except each field contains a value of S_i. The last field must contain the additional value S_{m+1}. Punch decimal point.
Set #6	1 card Cols. 1-4 Cols. 5-8	FORMAT (2I4) Value of M = number of entry lengths to be tried. 1 ≤ M ≤ 20. If more than 20 desired, repeat sets 6, 7 and 8. Right-justified with no decimal. Value of L = number of multipliers of F to be tried. 1 ≤ L ≤ 8. If more than 8 desired, repeat sets 6, 7 and 8. Right-justified with no decimal.
Set #7	1 card Cols. 1-80	FORMAT (8F10.0) This card has 8 fields of 10 columns each. Each field used contains a multiplier of F. Punch decimal point.
Set #8	1 ≤ cards ≤ 3 Cols. 1-80	FORMAT (8F10.0) This card has 8 fields of 10 columns each. Each field used contains an entry length. Punch decimal point.
Set #9	1 card Cols. 3-4	FORMAT (2I4) 99. Indicates the end of data set.

Sets 6, 7 and 8 can be repeated, as a group, until all values of entry length and multipliers have been tried, and must be followed by set 9. Sets 1 through 9 may be repeated.

* After last data set only, 1 card, FORMAT (5F10.0, I4), cols. 51-54: 9999. To indicate there is no more execution to perform.

No blank cards in any set.

No error messages.

Storage requirements are about 15,000 words, but this can be easily changed.

Special notes

Storage may be reduced by lowering the dimensions of variables dimensioned 1000 to the maximum number of time intervals needed. This program contains a multiple replacement statement at 60 + 1 and an end-of-file check on the standard input unit at 13 + 1. A replacement must be provided for the end-of-file check for the program to function properly. The statements READ N, list and PRINT N, list read and write on the standard input and output units.

Function statements EXPF(X) = EXP(X) and LOGF(X) = ALOG(X) must be inserted after the dimension statement for universal acceptance to FORTRAN IV compilers. A two-branch IF statement located at 50 + 2 should be changed to:

```
IF (AA(I).LE. S(J)) GO TO 17
```

for FORTRAN IV conformance.

YIELD 2

```
      PROGRAM YIELD2 (INPUT,OUTPUT,TAPE50=INPUT)
      DIMENSION F(1000),FM(1000),T(1000),S(1000),W(1000)
     1,X(1000),Y(1000),C(1000),XF(1000),          AA(25),B(8),U(8),
     2TITLE(10),V(8,25)
  100 READ 60, (TITLE(I),I=1,10)
      PRINT60,(TITLE(I),I=1,10)
   60 FORMAT(10A8)
      TF=TM=TT=TC=TW=TY=0.
      READ 2,POP,FL,FK,A,H,N
    2 FORMAT (5F10.0,I4)
      IF(N.EQ.9999) GO TO 101
      NN=N+1
    3 FORMAT(9F8.0)
      READ 3,(F(I),I=1,N)
      READ 3,(FM(I),I=1,N)
      READ 3,(S(I),I=1,NN)
   63 X(1)=POP
      DO 4 I=1,N
      G=(S(I+1)-FL)/(S(I)-FL)
      T(I)=-LOGF(G)/FK
      Z=F(I)+FM(I)
      E=F(I)*(1.-EXPF(-T(I)*Z))/Z
      W(I)=(A*(S(I)**H+S(I+1)**H))/2.
      C(I)=X(I)*E
      X(I+1)=X(I)*EXP(-Z*T(I))
   71 Y(I)=C(I)*W(I)
      TF=TF+F(I)*T(I)
      TM=TM+FM(I)*T(I)
      TT=TT+T(I)
      TC=TC+C(I)
    4 TY=TY+Y(I)
      TW=TY/TC
      PRINT 72
   72 FORMAT(1H0,55X,7HTABLE 1)
      PRINT 5
    5 FORMAT (1H0,45X,26HYIELD FOR KNOWN CONDITIONS/
     11H0,2X4HTIME,7X6HINTIAL,7X6HINTIAL,7X11HINST. FISH.,5X10HINST. NAT
     2.,5X4HTIME,6X9HESTIMATED,6X7HAVERAGE,7X9HESTIMATED/
     3  9H INTERVAL,5X6HLENGTH,5X10HPOPULATION,8X5HMORT.,10X5HMORT.,8X,
     45HLAPSE,5X11HCATCH (NO.),5X6HWEIGHT,6X11HYIELD (WT.)/
     51H 3X1HI,10X4HS(I),9X4HN(I),12X4HF(I),10X4HM(I),9X4HT(I),9X4HC(I),
     610X4HW(I),11X4HY(I)/1H0,119(1H*))
      DO 6 I=1,N
    6 PRINT 7,I,S(I),X(I),F(I),FM(I),T(I),C(I),W(I),Y(I)
      PRINT 8,TF,TM,TT,TC,TW,TY
    7 FORMAT(1H0,I4,F14.3,F13.1,F16.4,F14.4,F13.4,F13.2,F14.4,F17.4)
    8 FORMAT(1H0,6HTOTALS,25X,F16.4,F14.4,F13.4,F13.2,F14.4,F17.4)
   10 FORMAT(8F10.0)
   99 FORMAT(2I4)
   13 READ 99,M,L
      IF(M.EQ.99) GO TO 11
   12 READ 10,(B(I),I=1,L)
      READ 10,(AA(I),I=1,M)
      PRINT60,(TITLE(I),I=1,10)
      PRINT 9,(B(I),I=1,L)
```

YIELD 2

```
  9 FORMAT (1H1,50X,7HTABLE 2/1H0,31X,45HYIELDS (WT.) FROM TRIAL FISHI
    1NG PATTERNS WITH/
    21H ,32X42HVARYING ENTRY LENGTH AND FISHING INTENSITY/
    31H0,6HLENGTH2X,1H*,28X31HMULTIPLIER OF FISHING MORTALITY/
    41H ,2X2HAT,4X,1H*/
    51H ,5HENTRY3X,1H*,8F13.3/1H ,114(1H*))
      DO 40 I=1,25
      DO 40 J=1,8
 40   V(J,I)=0.
      DO 14 I=1,M
      DO 50 J=1,L
 50   U(J)=0.
      DO 15 J=1,N
      IF(AA(I).GT.S(J)) 16,17
 16   XF(J)=0.
      GOTO 15
 17   XF(J)=F(J)
 15   CONTINUE
      DO 18 J=1,L
      X(1)=POP
      DO 19 K=1,N
      XXF=B(J)*XF(K)
      Z=XXF+FM(K)
      E=XXF*(1.-EXPF(-T(K)*Z))/Z
      C(K)=X(K)*E
      V(J,I)=V(J,I)+C(K)
      X(K+1)=X(K)*EXP(-Z*T(K))
 19   U(J)=U(J)+C(K)*W(K)
 18   CONTINUE
 14   PRINT 20,AA(I),(U(J),J=1,L)
      PRINT 60,(TITLE(I),I=1,10)
      PRINT 89,(B(I),I=1,L)
 89   FORMAT (1H0,50X,7HTABLE 3/1H0,31X,45HCATCH (NO.)  FROM TRIAL FISHI
    1NG PATTERNS WITH/
    21H ,32X42HVARYING ENTRY LENGTH AND FISHING INTENSITY/
    31H0,6HLENGTH2X,1H*,28X31HMULTIPLIER OF FISHING MORTALITY/
    41H ,2X2HAT,4X,1H*/
    51H ,5HENTRY3X,1H*,8F13.3/1H ,114(1H*))
      DO 88 I=1,M
 88   PRINT 20,AA(I),(V(J,I),J=1,L)
 20   FORMAT(1H ,8X,1H*/1H ,F6.3,2X,1H*,8F13.3)
      GOTO 13
 11   N=N-1
      PRINT60,(TITLE(I),I=1,10)
      PRINT 92
 92   FORMAT (1H0,12X,7HTABLE 4/1H0,31HPOPULATION SIZE WITH NO HARVEST/
    11H0,2(8X,10HPOPULATION)/1H ,6HLENGTH,5X,7HNUMBERS,11X6HWEIGHT/
    21H0,36(1H*))
      X(1)=POP*EXPF(-T(1)*FM(1)/2.)
      DO 90 I=1,N
      YY=X(I)*W(I)
      X(I+1)=X(I)*EXPF((-T(I)*FM(I)-T(I+1)*FM(I+1))/2.)
      BA=(S(I+1)+S(I))/2.
 90   PRINT 91,BA,X(I),YY
 91   FORMAT (1H0,F5.2,F10.1,F18.2)
      GO TO 100
101   STOP
      END
```

Program name - GENPRD 2

Purpose

Fits the generalized stock production model described by Pella and Tomlinson (1969) to catch and effort data. This model estimates equilibrium yield as a function of effort or population size. The production curve is allowed to be skewed.

Programmer - Jerome J. Pella and Patrick K. Tomlinson
Modified by Catherine L. Berude

Originating organization - Inter-American Tropical Tuna Commission and
California Department of Fish and Game

Approximate storage - 12,300 (CDC 6600)

The original program is listed in Pella and Tomlinson (1969) and users are urged to read that publication before utilizing the program. Those authors also recommended that IATTC Internal Report No. 5 be read by those planning to fit the model to short series of catch and effort data. The listing contained here is a modified version of the original program relative to the arrangement of the data deck and the output format; computationally it is identical to the original.

The program consists of a main program (GENPRD2), a subroutine (MIN2), and a function (FOX).

The input card deck supplied by the user is defined in units:

Unit 1: One card, read by FORMAT (10A8)

Columns 1-80	Alphabetic title information centered on column 40. Species, effort unit, and catch unit identification may be included.

Unit 2: One card, read by FORMAT (I5)

Columns 1-5	The number of catch intervals (ND), right-justified without decimal.

Unit 3: ND cards, one for each catch interval ($1 < ND \leq 1000$), read by FORMAT (4F10.0/(3F10.0))

Columns 1-10	Catch obtained during the interval, left-justified with decimal punched.
Columns 11-20	Effort expended during the interval, left-justified with decimal punched.
Columns 21-30	Length of the interval, left-justified with decimal punched.
Columns 31-40	Time when interval began, left-justified with decimal punched. Time is entered in the form YYYY.fff where YYYY is the four-digit year and fff is the decimal fraction of the year preceding the beginning of the interval. For example, if the interval began on 1 October 1954, the field would contain 1954.750. This field is read from the first card only; it is used as t_0 in labelling output time intervals. On all cards, the field is optional; if it is blank on card 1, output intervals will begin at $t_0 = 0$. It provides a means for unique identification of catch cards; its presence on all cards allows convenient alteration of time range while maintaining interval labelling.

Unit 4: One card, read by FORMAT (215,4F10.0)

Columns 1-5	The number of subintervals desired for each time interval, right-justified without decimal. Try 1.
Columns 6-10	The number of times the step values are to be divided by 10, right-justified without decimal. The larger this number, the greater accuracy will be obtained; but the execution time will also increase. Try 3.
Columns 11-20	Starting step value for Fopt, left-justified with decimal punched. Note: If this field is zero or left blank, the initial step value will be taken as the "best guess" value input on the Fopt parameter card.
Columns 21-30	Starting step value for Q, left-justified with decimal punched. The note for columns 11-20 applies here as well.
Columns 31-40	Starting step value for r, left-justified with decimal punched. See note for columns 11-20.
Columns 41-50	Starting step value for Umax, left-justified with decimal punched. See note for columns 11-20.

Unit 5: Four cards, one for each parameter Fopt, Q, r, and Umax, respectively; each read by FORMAT (3F10.0)

Columns 1-10	"Best guess" of parameter value, left-justified with decimal punched.
Columns 11-20	Lower limit allowable for the parameter, left-justified with decimal punched.
Columns 21-30	Upper limit allowable for the parameter, left-justified with decimal punched.

If a parameter is to be held constant during a run, each of the three fields on that parameter's card should be punched with the same number. Card 1 contains Fopt data; card 2, Q; card 3, r; and card 4, Umax.

Unit 6: One or two cards, read by FORMAT (I5,(15F5.0))

Card 1:

Columns 1-5	The number of trial values of m, NM, right-justified without decimal. NM must not be greater than 24.
Columns 6-10	The first trial value of m, left-justified with decimal punched.
Columns 11-15 . . . Columns 76-80	Punched like columns 6-10, for each additional m trial value 2 through NM.

Card 2:

Columns 1-5 Columns 6-10 . . . Columns 41-45	Punched like columns 6-10 of Card 1, trial values of m.

If more than one data set is to be run in the same job, repeat Units 1 through 6 for each additional set.

Error messages: Power function errors can occur without destroying the iteration process.

GENPRD 2

The following section is from Pella and Tomlinson (1969), pp. 438-39.

SUGGESTIONS FOR GUESSING PARAMETERS

Since guesses are required when using GENPRD2, some hints for evaluating these are appropriate. The general situation is depicted as one in which data (catch and effort) are distributed over a range of stock sizes, including the optimum. The technique suggested is to choose Fopt equal to the mean of the observed efforts; choose Umax equal to the maximum observed catch per unit of effort; choose Pmax equal to 4 times the maximum observed catch; set q equal to Umax divided by Pmax; and set r equal to 0.8. The lower bounds of Fopt and Umax are set at 1/10 of the guesses, and the upper bounds are set at 10 times the guesses. The bounds of q should be more liberal, say 1/100 and 100 times. The bounds of r are obviously 0 and 1. The values for the step intervals are simply set equal to the guesses. These guesses are based on the assumption that all Δt_i equal 1.

Of course, if it is known that the catch-effort data were obtained from a segment of the range of stock sizes, or some Δt_i not equal to 1, then the guessing process must be modified. If serious doubt exists, one should make guesses as suggested above and at the same time set very wide bounds and utilize relatively large step intervals for a quick search across the range. If any of the final estimates equal a bound, the data should be rerun with wider bounds if the user does not consider solutions beyond the bounds as unfeasible. As for the values of m, it is appropriate to begin the search by choosing a range of values greater than 0 but less than 4 (m = 1 must be excluded). If little is known of the shape of the production curve, try 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, and 3.2 for a first run, then try additional values when the approximate range is determined by examining the values of S.

Several factors will influence the amount of computer time necessary to find a solution. The two most important are the values of N, the number of time subintervals, and KK, the number of subdivisions of step intervals, since each of these automatically increases the amount of computation as they become larger. Doubling N will roughly double computation. Increasing KK decreases the rate at which the search is made; and unless the search is near the estimate, convergence can be very slow. It would be advisable on initial runs to set N = 1, KK = 2 and use values of the step intervals which will search across the range with few steps. It should be pointed out that if independent information exists about one or more of the parameters, this can be incorporated by setting very narrow bounds to decrease computation time.

Special notes

Storage requirements can be reduced by lowering the dimensions of those variables dimensioned 1000 in the main program and subroutine FOX. They may be lowered to the maximum number of catch intervals (ND) which will be used. READ N, list and PRINT N, list are used to read and write from the standard input and output units. An end-of-file check on the standard input unit occurs at statement 11+1. The program should operate satisfactorily if this statement is removed. The statement function "EXPF(X) = EXP(X)" should be inserted after the dimension statement of the main program. For FORTRAN IV conformity the following three two-branch IF statements should be changed.

Statement location	Revised statement
103 + 8	IF (T(I)) 13,40,13
3 + 2	IF (E(J)) 14,15,14
30 + 1	IF (T(J)) 17,18,17.

Reference

Pella, Jerome J. and Patrick K. Tomlinson, A generalized stock production model. Inter-Amer. 1969 Trop.Tuna Comm.,Bull., 13(3):419-496.

GENPRD 2

```
      PROGRAM GENPRD2 (INPUT,OUTPUT,TAPE50=INPUT)
      DIMENSION  C(1000),E(1000),T(1000),X(4),BL(4),UL(4),G(4),D(4),
     1 XM(24),SP(10)
      COMMON/STUT/C,E,T,ND,XM,I,N
C
C        PROGRAM GENPRD2 8/70 INCORPORATES INPUT CHANGES, CALCULATION
C        OF AN ADDITIONAL VALUE = OPTIMUM CATCH PER UNIT OF EFFORT, AND
C        OUTPUT REARRANGEMENT INCLUDING PRINT OF INPUT LOWER AND UPPER
C        LIMITS ON CURVE PARAMETERS FOPT, LITTLE R, Q, AND UMAX.
C        STORAGE FOR THE LOWER AND UPPER LIMITS IS CHANGED FROM THE
C        SINGLE VECTOR A TO TWO VECTORS BL AND UL (SUBR MIN CHANGED TOO)
C
C       SP(*) CONTAINS TITLE DATA
      NTHRU=0
 5    READ 11, (SP(J),J=1,10)
 11   FORMAT(10A8)
      IF(EOF,50)8,9
C       ND = NO. OF CATCH INTERVALS, YRINIT = YEAR OF FIRST TIME PERIOD,
C       C(*), E(*), T(*) = CATCHES, EFFORTS, LENGTHS OF INTERVAL FOR
C       EACH CATCH INTERVAL (ONE CARD PER INTERVAL, MAX=1000)
    9 READ 100, ND,C(1),E(1),T(1),YRINIT,(C(I),E(I),T(I),I=2,ND)
  100 FORMAT (I5/4F10.0/(3F10.0))
      TZERO=YRINIT
C       N = NO. OF SUBINTERVALS DESIRED FOR EACH TIME INTERVAL,
C       KK = NO. OF TIMES STEP INTERVALS, D(*), ARE TO BE DIVIDED BY 10,
C       D(*) = STARTING STEP VALUES FOR FOPT, Q, LITTLE R, UMAX (IF ZERO
C       OR BLANK, ARE INITIALLY SET TO INPUT GUESSES OF THOSE PARAMS.)
      READ 101, N,KK,(D(I),I=1,4)
  101 FORMAT (2I5,4F10.0)
C       FOLLOWING CARD INPUTS CALCULATION PRECISION CONTROLS.
C       FOLLOWING READ-ONE CARD CONTAINING PARAMETER ESTIMATE DATA FOR
C       EACH FOPT, Q, LITTLE R, UMAX, RESPECTIVELY.
C       G(*) = BEST GUESS OF PARAMETER VALUE, BL(*) = LOWER LIMIT,
C       UL(*) = UPPER LIMIT
      READ 102, (G(I),BL(I),UL(I),I=1,4)
  102 FORMAT (3F10.0)
C       FOLLOWING READ INPUTS LITTLE M TRIAL VALUES (MAX=24)
C       NM = NO. OF TRIAL VALUES, XM(*) = TRIAL VALUES OF LITTLE M
      READ 103, NM,(XM(I),I=1,NM)
  103 FORMAT (I5,(15F5.0))
      NV=4
      FN=N
      SQ=0.
      SUM=0.
      FND=ND
      DO 40 I=1,ND
      SUM=SUM+C(I)
      IF(T(I)) 13,40
   13 E(I)=E(I)/T(I)
   40 T(I)=T(I)/FN
      AV=SUM/FND
      DO 42 I=1,ND
   42 SQ=SQ+(C(I)-AV)**2
      T(ND+1)=1.
      DO 20 K=1,NM
      I=K
      Z=1./(XM(I)-1.)
      CALL MIN2(NV,KK,D,BL,UL,G,X,S)
      R=(SQ-S)/SQ
      POPT=X(4)/(XM(I)**Z*X(2))
      XK=XM(I)*X(1)*X(2)/(1.-XM(I))
      XH=X(2)**(XM(I)-1.)*X(4)**(1.-XM(I))*XK
      CMAX=X(1)*X(2)*POPT
      PMAX=(XK/XH)**(1./(XM(I)-1.))
      P0=X(3)*PMAX
      UOPT=CMAX/X(1)
      PRINT 200, (SP(J),J=1,10),XM(I)
```

```
  200 FORMAT (1H1,20X,10A8//41X,34HGENERALIZED PRODUCTION MODEL   M =,
     1F5.2)
      PRINT 201, CMAX,X(1),UOPT,POPT
  201 FORMAT (//2X,126(1H*)//12X,22HMAX. EQUILIBRIUM CATCH,11X,14HOPTIMU
     1M EFFORT,13X,17HOPT. CATCH/EFFORT,12X,18HOPTIMUM POPULATION/20X,
     2 6H(CMAX),23X,6H(FOPT),23X,6H(UOPT),23X,6H(POPT)//4E29.8)
      PRINT 202, X(2),X(4),X(3),BL(2),BL(1),BL(4),BL(3),UL(2),UL(1),
     A UL(4),UL(3)
  202 FORMAT (//2X,126(1H*)//13X,19HCATCHABILITY COEFF.,19X,1H*,20X,17HM
     1AX. CATCH/EFFORT,10X,22HINITIAL POP./MAX. POP./21X,3H(Q),25X,6H(FO
     2PT),23X,6H(UMAX),20X,11H(P(0)/PMAX)//E29.8,29X,2E29.8//10X,11HLOWE
     3R LIMIT,E15.7,3(9X,2HLL,E18.7)//10X,11HUPPER LIMIT,E15.7,3(9X,2HUL
     4,E18.7) )
      PRINT 203, PMAX,N,KK,XH,XK
  203 FORMAT (//2X,126(1H*)//44X,20HCOMPUTING PARAMETERS,25X,18HEQUATION
     1 CONSTANTS/13X,15HMAX. POPULATION,6X,17HTIME SUBINTERVALS,8X,13HST
     2EP INTERVAL,11X,8HCONSTANT,14X,8HCONSTANT/18X,6H(PMAX),17X,3H(N),
     3 13X,17HSUBDIVISIONS (KK),12X,3H(H),19X,3H(K)//E28.8,I15,I21,6X,
     4 2E22.6)
      PRINT 204, S,R,SUM,AV
  204 FORMAT (//2X,126(1H*)//16X,14HSUM OF SQUARES,21X,1HR,24X,11HTOTAL
     1CATCH,17X,13HAVERAGE CATCH//4E29.7///2X,126(1H*)///2X,126(1H*)/
     2 21X,1H*,17X,1H*/9X,4HTIME,8X,1H*,4X, 9HAT END OF,4X,1H*,37X,15HDU
     3RING INTERVAL/7X,8HINTERVAL,6X,1H*,5X,8HINTERVAL,4X,1H*/21X,1H*,
     417X,1H*,6X,7HAPPLIED,10X,8HOBSERVED,10X,8HEXPECTED,10X,8HOBSERVED,
     510X,8HEXPECTED/4X,5HBEGIN,6X,3HEND,3X,19H* POPULATION SIZE *,7X,
     6 6HEFFORT,12X,5HCATCH,13X,5HCATCH,9X,12HCATCH/EFFORT,6X,12HCATCH/E
     7FFORT/2X,126(1H*))
      TT=T(1)*FN + YRINIT
      DO 20 J=1,ND
      PCAT=0.
      DO 3 L=1,N
      P1=(XH/(XK+X(2)*E(J))+(P0**(1.-XM(I))-XH/
     1 (XK+X(2)*E(J)))*EXPF(-(XK+X(2)*E(J))*(1.-XM(I))
     2  *T(J)))**(1./(1.-XM(I)))
      PCAT=PCAT+X(2)*E(J)*T(J)*(P0+P1)/2.
3     P0=P1
      E(J)=E(J)*T(J)*FN
      IF(E(J)) 14,15
   14 U=C(J)/E(J)
      EU=PCAT/E(J)
      GOTO 16
   15 U=0.
      EU=0.
   16 PRINT 30, YRINIT,TT,E(J),C(J),PCAT,U,EU,TT,P1
   30 FORMAT (/2F10.3,18X,5E18.7/10X,F10.3,E18.7)
      IF(T(J)) 17,18
   17 E(J)=E(J)/(T(J)*FN)
   18 YRINIT=TT
      TT=TT+T(J+1)*FN
20    CONTINUE
      GO TO 5
8     STOP
      END
```

```
      FUNCTION FOX(X)
      COMMON/STOT/C,E,T,ND,XM,I,N
      DIMENSION E(1000),C(1000),X(4),T(1000),XM(24)
      FOX=0.
      FN=N
      P0=X(3)*X(4)/X(2)
      XK=XM(I)*X(1)*X(2)/(1.-XM(I))
      XH=X(2)**(XM(I)-1.)*X(4)**(1.-XM(I))*XK
      DO 5 J=1,ND
      PCAT=0.
      DO 3 K=1,N
      IF(P0-0.) 6,6,7
7     XX=1.-XM(I)
      Z=XK+X(2)*E(J)
      IF(Z.EQ.0.) Z=.0000001
      ZZ=-Z*XX*T(J)
      IF(ABS(ZZ).GT.741.) GO TO 6
      P1=(XH/Z +(P0**XX-XH/Z)*EXP(ZZ))**(1./XX)
      PCAT=PCAT+X(2)*E(J)*T(J)*(P0+P1)/2.
3     P0=P1
5     FOX=FOX+(PCAT-C(J))**2
      RETURN
6     FOX=1.E+50
      RETURN
      END
```

```
      SUBROUTINE MIN2 (NV,KK,DEL,BL,UL,GUESS,X,FOFX)
      DIMENSION BL(1),UL(1),GUESS(1),XNEW(50),XNOW(50),X(1),DEL(1)
      DIMENSION DELTA(50)
      NK = KK
      NX = NV
      DO 5 I = 1, NX
      XNOW(I) = GUESS(I)
      XNEW(I) = XNOW(I)
    5 DELTA(I)=DEL(I)
      DO 14 J=1,NX
      IF(DELTA(J)) 11,11,14
   11 DELTA(J)=GUESS(J)
14    CONTINUE
20    FNOW = FOX(XNOW)
201   FOLD = FNOW
200   DO 40 I = 1, NX
      XNEW(I) = XNEW(I) + DELTA(I)
      NA = NX+I
      IF(XNEW(I)-UL(I)) 22,22,21
   21 XNEW(I)=UL(I)
22    FNEW = FOX(XNEW)
      IF(FNEW - FNOW) 25, 26, 26
25    FNOW = FNEW
      GO TO 40
26    XNEW(I) = XNOW(I) - DELTA(I)
      IF(XNEW(I)-BL(I)) 28,30,30
   28 XNEW(I)=BL(I)
```

```
30    FNEW = FOX(XNEW)
      IF(FNEW - FNOW) 25, 34, 34
34    XNEW(I) = XNOW(I)
40    CONTINUE
      IF(FNOW - FOLD) 50, 45, 45
45    NK = NK - 1
      IF(NK) 46, 46, 47
  47  DO 48 J=1,NX
  48  DELTA(J)=DELTA(J)*.1
      GO TO 200
50    DO 60 I = 1, NX
      T = XNOW(I)
      XNOW(I) = XNEW(I)
      XNEW(I) = 2.*XNEW(I) - T
      NA = NX+I
      IF(XNEW(I)-UL(I)) 52,60,51
  51  XNEW(I)=UL(I)
      GO TO 60
  52  IF(XNEW(I)-BL(I)) 53,60,60
  53  XNEW(I)=BL(I)
60    CONTINUE
      FNEW = FOX(XNEW)
      IF(FNEW - FNOW) 65, 70, 70
65    FNOW = FNEW
      GO TO 50
70    DO 71 I = 1, NX
71    XNEW(I) = XNOW(I)
      GO TO 201
 46   FOFX = FNOW
      DO 80 I = 1, NX
80    X(I) = XNOW(I)
      RETURN
      END
```

Program name - POPSIM

Purpose

By means of population simulation and yield optimization this program provides a general framework for evaluating long-term yields from populations having density-dependent recruitment, and for seeking maximum long-term yield as a function of time-specific fishing rates.

Programmer - Carl J. Walters

Originating organization - Institute of Animal Resource Ecology, University of British Columbia

Approximate storage - 3,000 (Data storage only)

Structure

The program has two basic components: (1) a population generator which calculates age structure, reproduction, and yield; and (2) an optimizing routine which controls input to the population generator. A "yield trial" in the program consists of a single n-year simultation of the population. The n years are divided into k periods of equal length (L years), and each period is assigned an overall fishing rate. The fishing rate for each L-year period is further divided into age-specific and (under one option) within-year specific components. The optimization routine treats n-year yield as a response surface on the k period-specific fishing rates, and searches this surface for best n-year yield by repeatedly simulating the population at different values for the k fishing rates.

Options

The optimization routine can perform: (1) a steepest-ascent response surface search, or (2) repeated population simulation at systematically increasing levels of fishing intensity.

Growth and mortality can be specified in two ways: (1) "Bertalanffy-annual" in which growth is assumed to follow the Bertalanffy growth function:

$$w_t = w_\infty (1-e^{-kt})^3$$

and age specific natural and fishing mortality rates are assumed constant within each year. Age-specific mortality rates need not be equal. Under this option, yield is calculated for each age group on each year by a modification of the Beverton-Holt yield equation. Natural mortality rates are entered as instantaneous annual rates.

(2) "linear-segmental", in which the growth curve is specified by a series of linear segments for each age group within each simulated year. Age-specific fishing and natural mortality rates can be varied within each age, but are assumed constant over each time interval defined by a segment of the growth curve. Linear growth coefficients for the program are analagous to "instantaneous growth rates" of the Ricker yield model, but should be calculated for the i^{th} age group, j^{th} within-year interval as:

$$g_{ij} = \frac{w_{ije} - w_{ijo}}{\Delta_j t}$$

where w_{ijo} is weight at the start of the j^{th} interval, w_{ije} is weight at the end of the interval, and $\Delta_j t_{ijo}$ is length of the j^{th} interval. The analagous instantaneous growth rate would be:

$$G_{ij} = \frac{1}{\Delta_j t} \log_e(w_{ije}/w_{ijo}).$$

Again, natural mortality rates for each age group and interval are entered as instantaenous annual rates, while intervals to receive fishing age specified by lower and upper j values.

The stock-reproduction function is specified by the user through a subroutine (EGPROD). Time between birth and recruitment into the simulated population can be varied from 0 to 10 years.

Calculation of fishing intensity

Fishing intensity is represented in the model by instantaneous annual harvest rates. Under the Bertalanffy-annual option, fishing rates for each age group on each year (y) of a simulation period are calculated from the overall fishing intensity for the period, F_p, as:

$$F_{iy} = 0 \text{ if } i < i_{min} \text{ (minimum age fished)}$$
$$F_{iy} = F_p \cdot \mu_{i-imin+1} \text{ otherwise,}$$

where $\mu_{i-imin+1}$ is a "mesh-selection multiplier" specifying relative harvest intensity for age group i as a proportion of maximum harvest intensity.

Under the segmented growth option,

$$F_{ij} = 0 \text{ if } i < i_{min}$$
$$F_{ij} = F_p \cdot \mu_{i-imin+1} \cdot \gamma_j \text{ otherwise}$$

where
$$\gamma_j = \begin{cases} 1 \text{ if } j_{min} \leq j \leq j_{max} \\ 0 \text{ otherwise} \end{cases}$$

So, the segmented growth option assumes that harvest rate will be constant over within-year intervals j_{min} through j_{max}, and will be 0 at other times during the year.

Simulation procedure

Basic equations of the population generation model are described in Walters (1969). Calculations for each year in a yield trial are made in the following sequence:

(1) Apply mortality to initial age structure for the year (input-specified for first year of every trial, otherwise calculated) to create age structure at spawning time; "initial age structure" for each year is assumed to be the number in each age group at the time young are entered into the population.

(2) Call subroutine EGPROD which must return the number of young to be entered into the population (as simulated age group "1") at the start of the next simulated year.

(3) Store age structure at spawning time as first row of a matrix which carries age structures at spawning time for years prior to the one under consideration, and advance age-structures already stored in the matrix down one row; if first year in the yield trial, set all prior spawning time age structures to spawning time age structure for the first year.

(4) Calculate yield by age group for the year.

(5) Calculate number in each age group at the end of the year, and advance age of animals to form initial age structure for next year.

Steps (1) to (5) are repeated n times (k.L times) for each yield trial.

Exponential functions for growth, mortality, and yield are evaluated only at the start of each L-year period rather than every year, in order to save computer time. Thus, natural mortality, growth, and fishing rates may not be varied within each period, but can be changed (if desired) from period to period by using statements in subroutine EGPROD.

Steepest-ascent optimization procedure

Under the response surface search option, the k-variate gradient vector $V_s = \{\partial Y_s / \partial F_p\}$ is evaluated approximately after each (s^{th}) yield trial as $V_{sp} = \frac{\Delta Y_s}{\Delta F_p}$,

where Y_s is total n-year yield for the trial, and F_p is fishing rate during the p^{th} period in the trial. The approximate gradient vector is found by determining Y'_s at $F'_p = F_p + d$ where d is small and $p = 1, \ldots, k$. Thus the population is actually simulated $k+1$ times (over n years each time) at each iteration in the steepest ascent search. The F_p for the $s+1^{st}$ iteration are chosen as:

$$F_{p\,(s+1)} = F_{p\,(s)} + (\tilde{\nabla}^*_s)_p \delta_s$$

where $\tilde{\nabla}^*_s$ is a normed form of the approximate gradient vector:

$$(\tilde{\nabla}^*_s)_p = \left(\frac{\Delta Y_s}{\Delta F_p}\right) \Big/ \left\{\sum_i^k \left(\frac{\Delta Y_s}{\Delta F_i}\right)^2\right\}^{1/2}$$

The F_p are constrained to be greater than or equal to 0. If at any iteration $F_p = 0$ and $(\tilde{\nabla}^*_s)_p < 0$, then $(\nabla_s)_p$ is set to 0. δ_s is a maximum search increment, partly controlled by the user, such that:

(a) $\delta_{s+1} \leq \delta_s$ and $\delta_{s+1} < \delta_s$ only if $Y_{s+1} \leq Y_s$

(b) $\sum_{s=1}^{\infty} \delta_s = \infty$

(c) $\sum_{s=1}^{\infty} \delta_s^u < \infty$ for all $u > 1$

The geometric series (1, 1/2, 1/3, ...) is the strongest which must always force convergence to a maximum, but the user may find it necessary to experiment with other convergence criteria to find the best one for any problem.

Search is discontinued when either all gradient vector elements or all increments in the F_p fall below minima specified by the user.

Care must be taken in using the steepest ascent search procedure, because the response surface $Y_s = f\{F_1, \ldots, F_k\}$ is strongly affected by: (1) initial age structure; (2) the size of n relative to m, the maximum number of age groups present at any time; and (3) the initial trial values of F_p. If n is small the model may predict that maximum yield would be obtained by wiping out the population. So, n should be made large (greater than 30) when using the ascent search option. The program will usually predict, unless L is very large, that fishing rate should be made very large in the k^{th} period. Also, the response surface may have multiple local maxima if n is small while k is greater than 2. The local maxima occur at points like $F = (0, \ldots, 0, V_p, V_{p+1}, \ldots, V_k)$ where V_p is large and $p > 1$. A suggested way to avoid these local maxima is to take as an initial estimate of the vector $F = (v, v, \ldots, v)$ where $v \simeq 0.5$.

Formulation of problems

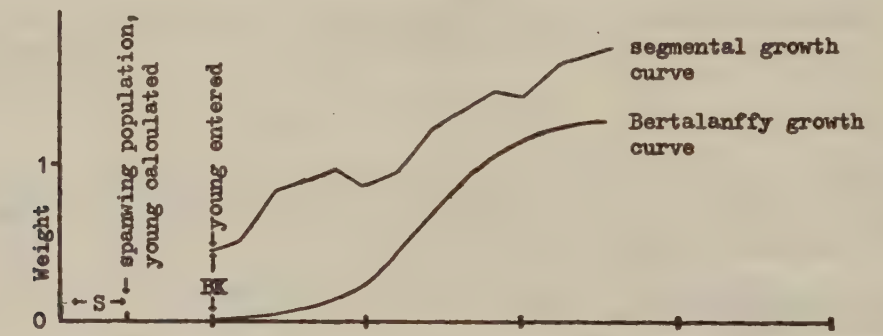
Selection of starting times and time scale for computer simulation is arbitrary, and may be confusing to the user. As a convention, this model begins each simulated year at the time young are to be "entered" into the population. Entry of young consists simply in adding another year class to the simulated population. The new year class is designated at age group "1", regardless of true age of fish entered.

As a second convention, the model requires that the user calculate for any year the number of "young" to be entered into the population at the start of the next year, in terms of some previous year's spawning population. The model could just as well have been structured to have the user calculate the number of young to be added after x years from the spawning time of the year under consideration.

Growth and mortality rate intervals within the typical simulated year must be ordered on a 0-1 time scale with 0 representing the time when young are to be entered into the population. Also, growth and mortality rates must be re-indexed from true to simulated age. For example, if the time lag from birth to simulated entry into the population were three years, then mortality and growth rates entered for simulated age group one would have to correspond to true age group 4, and so forth. The Bertalanffy growth curve is automatically adjusted to true age, but segmental growth curve estimates must be adjusted to true age by the user.

Two graphical examples are presented below to show for a typical cohort how the program indexes age groups and applies growth for different birth-entry time lags. Arbitrary numbers are used for clarity; the graphic-tabular layout used may prove useful in organizing data for simulation. Input variables "S" and "LAG" are used to generate the birth-entry lag, where the absolute time from birth to entry is LAG+S-1 years. BK is (1) the individual weight of fish at the time they enter the simulated population (segmented growth option), or (2) exponent in the Bertalanffy growth curve. KKK is the index of a typical year.

Example 1: LAG=0 (cohort enters population 1-S years after "birth" from spawning run)



Simulated year	KKK	KKK+1	KKK+2	KKK+3	KKK+4	...
Simulated age	--	1	2	3	4	...
(True age)	(0)	(0-1)	(1-2)	(2-3)	(3-4)	...

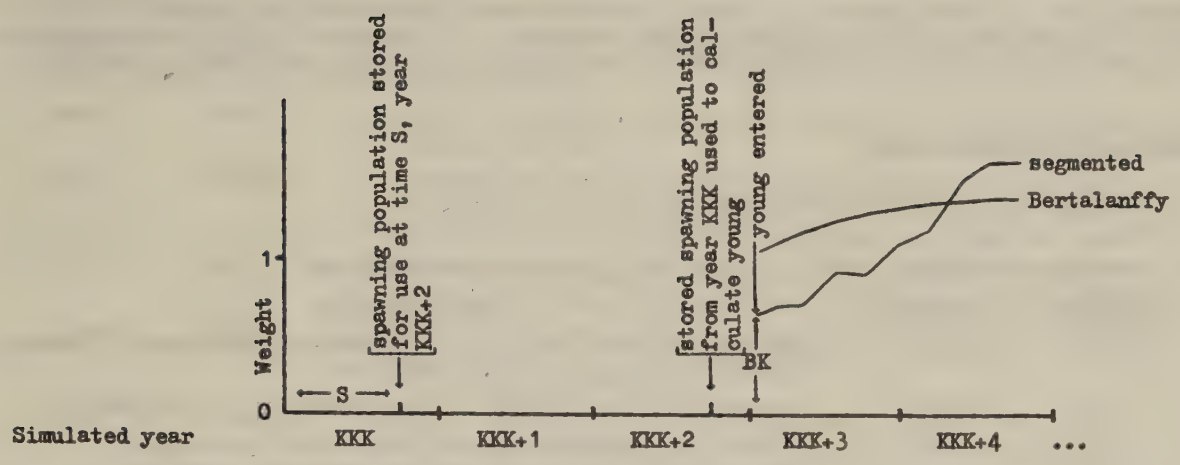
Growth coefficients (segmental option)

Interval						
1	0.5	0.5	.	.	.	
2	1.5	1.0	.	.	.	
3	0.25	.	.	.	.	
4	-0.5	.	.	.	.	

Mortality coefficients (same option)

Interval						
1	0.9	0.4	.	.	.	
2	0.8	0.5	.	.	.	
3	0.6	.	.	.	.	
4	0.5	.	.	.	.	

Example 2: LAG-3



Simulated year	KKK	KKK+1	KKK+2	KKK+3	KKK+4	...
Simulated age	--	--	--	1	2	...
(True age)	(0)	(0-1)	(1-2)	(2-3)	(3-4)	...

Growth coefficients

..... As example 1 .....

Mortality coefficients

..... As example 1 .....

Note that the within-year intervals under the segmented growth option must be the same for all age groups. "Fishing seasons" may be set only by varying interval boundaries, since it is assumed that a set of intervals, not time points, defines the fishing season. Thus, more within-year intervals may need to be defined than would be necessary to describe growth and mortality for any single age group.

The stock-reproduction relationship has two basic components: a measure of spawning stock size, and equation(s) specifying number of young as a function of the spawning stock measure. Provision is made in the program to read in one relative reproductive parameter for each age group (as age-specific fecundity or mean weight), so the spawning stock measure may be expressed as:

$$R = \sum_{i=1}^{m} a_i N_i(S)$$

where some of the a_i may be 0. Two additional parameters are read in to be used in specifying young = f(R), where f(R) is specified by subroutine.

Sensitivity analysis

An input option permits the user to evaluate sensitivity of long-term yield to changes in population parameters (growth, mortality, reproductive) under the Bertalanffy-annual option. The sensitivity analysis subroutine returns change in n-year yield produced by a one-percent change in each population parameter, evaluated at the best harvest regime (F_p set) encountered during the run.

INPUT AND RESTRICTIONS

Program input is of two types: (1) sets of cards to specify population parameters, initial harvest regime and age structure, and search procedure; (2) subroutine EGPROD, which specifies a reproductive function for all data sets in the run (the card input series may be repeated).

Control and parameter cards

1. TITLE CARD - alphameric identification in cols. 1-80

2. OPTIONS AND CONTROL CARD - (all I-fields, punched without decimal)

Columns	Variable	Identification
1	IOV	Search procedure option: 0 = systematic variation of fishing rate; 1 = steepest ascent search on F
2	IOG	Growth-mortality option: 0 = Bertalanffy; 1 = segmental
3	IOS	Sensitivity analysis option (Bertalanffy option only): 0 = no sensitivity; 1 = perform sensitivity analysis
4-6	IMAX	Maximum number of yield trials in the run ($1 \leq$ IMAX)
7-8	IOC	Number of yield trials between each full printout of age structure and yield; every IOCth yield trial will be accompanied by a full printout of age structure and yield during the trial ($1 \leq$ IOC)
9-10	M	Number of age groups to be carried ($1 \leq M \leq 10$)
11-12	K	Number of "planning periods" in each yield trial (see restriction below)
13-14	L	Length of each "planning period" in years ($K \times L \leq 50$)
15-16	IMIN	Youngest age group to receive fishing ($1 \leq$ IMIN $\leq M$)
17-18	LAG	Number of years between birth and recruitment into population (see above examples) ($0 \leq$ LAG ≤ 10)
19-20	IPER	Number of growth-mortality periods into which each year is to be divided, if IOG = 1 ($1 \leq$ IPER ≤ 10)
21-22	ITL	First period within year to receive fishing under segmental option ($1 \leq$ ITL ≤ 10)
23-24	ITH	Last period within each year to receive fishing under segmental option (ITL $\leq$ ITH ≤ 10)

3. PARAMETER CARD - (all fields punched with decimal point)

Columns	Variable	Identification
1-5	W	Asymptotic weight in Bertalanffy growth curve; not used in segmental option
6-10	BK	Growth coefficient ("k") in Bertalanffy growth curve, or weight of fish at entry to simulated population under segmental option
11-15	AR1	Parameter of reproduction curve
16-20	AR2	Parameter of reproduction curve
21-25	S	Spawning time, in years from time last young are entered into population ($0 < S < 1$)
26-30	PINC	Increment to be applied to each year's fishing rate for every yield trial after the first (used only for systematic search option)

3. PARAMETER CARD (Cont'd)

Columns	Variable	Identification
(the following fields may be left blank)		
31–40	DELTA	Maximum increment to be applied to any period's fishing rate during ascent search
41–50	PEWINC	Increment on DELTA's divisor in converging to maximum during ascent search (recommended PEWINC = 1., which generates geometric decrease in DELTA after convergence on maximum begins)
51–60	DEL	Magnitude of change to be applied to each period's fishing rate in evaluating gradient during ascent search (DEL≪DELTA)
61–70	GMIN	Ascent search will terminate when all elements in the fishing rate gradient vector fall below GMIN; a convergence criterion for maximum yield attainment
71–80	DELMIN	Ascent search will terminate when all changes to be applied to fishing rates for subsequent yield trial fall below DELMIN; again, a convergence criterion for maximum yield attainment

4. REPRODUCTIVE PARAMETERS CARD – (punched with decimal points)

Columns	Variable	Identification
1–5	F(1)	First reproductive parameter
6–10	F(2)	Second reproductive parameter
.	.	.
.	.	.
.	.	.
(5M–4)–5M	F(M)	M^{th} reproductive parameter
.	.	.
.	.	.
.	.	.
46–50	F(10)	Tenth reproductive parameter

5. INITIAL AGE STRUCTURE CARD – (punched with decimal points)

Columns	Variable	Identification
1–5	XI(1)	Number in age group 1 at start of year 1 (each trial)
6–10	XI(2)	Number in age group 2 at start of year 1 (each trial)
.	.	.
.	.	.
.	.	.
(5M–4)–5M	XI(M)	Number in age group M at start of year 1 (each trial)

6. FIRST-TRIAL FISHING RATES CARD – (punched with decimal points)

Columns	Variable	Identification
FIRST CARD: fishing rates for first 10 periods in yield trial 1		
1–5	PP(1)	Instantaneous fishing rate for first period
6–10	PP(2)	Instantaneous fishing rate for second period
.	.	.
.	.	.
.	.	.
46–50	PP(10)	Instantaneous fishing rate for tenth period

6. FIRST-TRIAL FISHING RATES CARD (Cont'd)

Columns	Variable	Identification
SECOND CARD:		
1-5	PP(11)	Instantaneous fishing rate for 11^{th} period
.	.	.
.	.	.
.	.	.
46-50	PP(20)	Instantaneous fishing rate for 20^{th} period

ETC. AS NEEDED TO READ IN K FISHING RATES AT 10 PER CARD

7. RELATIVE FISHING INTENSITY AT AGE CARD - (punched with decimal points)

Columns	Variable	Identification
1-5	PS(1)	Fishing intensity on age group IMIN as proportion of intensity on fully recruited ages
6-10	PS(2)	Fishing intensity on age group IMIN+1 as proportion of intensity on fully recruited ages
.	.	.
.	.	.
.	.	.
46-50	PS(10)	Fishing intensity on age group IMIN+10 as proportion of intensity on fully recruited ages

8. AGE-SPECIFIC NATURAL MORTALITY RATES (CARD INCLUDED ONLY IF BERTALANFFY GROWTH OPTION USED) - (values punched with decimal point)

Columns	Variable	Identification
1-5	Q(1)	Instantaneous natural mortality rate for age group 1
6-10	Q(2)	Instantaneous natural mortality rate for age group 2
.	.	.
.	.	.
.	.	.
(5M-4)-5M	Q(M)	Instantaneous natural mortality rate for age group M

FOLLOWING CARDS INCLUDED FOR SEGMENTED GROWTH OPTIONS ONLY;
CARD GROUPS 1-8 FILL INPUT REQUIREMENTS FOR BERTALANFFY OPTION

9. MORTALITY AND GROWTH INTERVAL LENGTHS - (punched with decimal point)
NOTE: mortality period lengths must sum to 1

Columns	Variable	Identification
1-5	TAU(1)	Length (years) of first within-year interval ($0 \leq TAU(I) \leq 1$)
6-10	TAU(2)	Length (years) of second within-year interval ($0 \leq TAU(I) \leq 1$)
.	.	.
.	.	.
.	.	.
(5xIPER-4)-5xIPER	TAU(IPER)	Length (years) of $IPER^{th}$ within-year interval ($0 \leq TAU(I) \leq 1$)

10. INTERVAL-AGE SPECIFIC NATURAL MORTALITY RATES CARD - (punched with decimal point)

M cards are necessary, one card for each age group; each card contains the within-year natural mortality rates (instantaneous) for an age group; thus there are IPER values per card. Mortality rates should be entered on an annual basis, though they will be applied for shorter time intervals.

10. NATURAL MORTALITY RATES CARD (Cont'd)

Columns	Variable	Identification
FIRST CARD: age group 1 mortality rates		
1-5	QW(1,1)	Instantaneous rate for within-year period 1
6-10	QW(2,1)	Instantaneous rate for within-year period 2
.	.	.
.	.	.
.	.	.
(5xIPER-4)-5xIPER	QW(IPER,1)	Instantaneous rate for within-year period IPER
SECOND CARD: age group 2 mortality rates		
1-5	QW(1,2)	Instantaneous rate for within-year period 1
6-10	QW(2,2)	Instantaneous rate for within-year period 2
.	.	.
.	.	.
.	.	.
(5xIPER-4)-5xIPER	QW(IPER,2)	Instantaneous rate for within-year period IPER
		.
		.
		.
M^{th} CARD: age group M natural mortality rates		
1-5	QW(1,M)	Instantaneous mortality rate for within-year period 1
6-10	QW(2,M)	Instantaneous mortality rate for within-year period 2
.	.	.
.	.	.
.	.	.
(5xIPER-4)-5xIPER	QW(IPER,M)	Instantaneous mortality rate for within-year period IPER

11. INTERVAL-AGE SPECIFIC LINEAR GROWTH COEFFICIENTS CARD

As in (10) above, M cards are necessary; each card contains the within-year linear growth coefficients for an age group; slopes should be measured in units per year, though they will be applied for shorter times.

Columns	Variable	Identification
FIRST CARD: age group 1 linear growth coefficients		
1-5	BW(1,1)	Slope of weight-time curve for interval 1 in year
6-10	BW(2,1)	Slope of weight-time curve for interval 2 in year
.	.	.
.	.	.
.	.	.
(5xIPER-4)-5xIPER	BW(IPER,1)	Slope of weight-time curve for interval IPER in year
SECOND CARD: age group 2 linear growth coefficients		
1-5	BW(1,2)	Slope of weight-time curve for interval 1 in year
6-10	BW(2,2)	Slope of weight-time curve for interval 2 in year
.	.	.
.	.	.
.	.	.
(5xIPER-4)-5xIPER	BW(IPER,2)	Slope of weight-time curve for interval IPER in year
		.
		.
		.

11. LINEAR GROWTH COEFFICIENTS CARD (Cont'd)

Columns	Variable	Identification
M^{th} CARD: age group M linear growth coefficients		
1-5	BW(1,M)	Slope of weight-time curve for interval 1 in year
6-10	BW(2,M)	Slope of weight-time curve for interval 2 in year
.	.	.
.	.	.
.	.	.
(5xIPER-4)-5xIPER	BW(IPER,M)	Slope of weight-time curve for interval IPER in year

Subroutine EGPROD

This subroutine is called at times S in every year in each simulation trial. It must return the number of young to be entered into the population as simulated age group 1 at the start of the next year, as the FORTRAN variable EGGS. A COMMON statement (provided with the program deck) supplies the subroutine with the following information:

KKK - the year under consideration, which varies from 1 to KxL within each yield trial

M - number of age groups in the population

W, BK, or BW(I,J) - growth coefficients

Q(I) or QW(I,J) - natural mortality rates

S - spawning time

AR1, AR2, F(1), ..., F(10) - reproductive parameters from input

XF(1), ..., XF(M) - number in each age group at the start of that year (at time last young entered into population)

XS(1), ..., XS(M) - number in each age group at time S within the year under consideration

SP(1,1), SP(1,2), ..., SP(1,M) - number in each age group at time S on previous year (KKK-1)

SP(2,1), SP(2,2), ..., SP(2,M) - number in each age group at time S on year KKK-2

.
.
.

SP(LAG,1), SP(LAG,2), ..., SP(LAG,M) - number in each age group at time S in year KKK-LAG (note that for KKK< LAG, these variables will contain spawning population for year 1)

The following example illustrates how these variables may be used to construct a Ricker stock-recruitment relation:

```
      SUBROUTINE EGPROD
            .
         [COMMON block as provided with deck]
            .
      E=0.
      DO 1 I=1,M
    1 E=E+SP(LAG,I)*F(I)
      EGGS=E*EXP(AR1*(1.-E*AR2))
      RETURN
      END
```

If one has information about recruitments likely to occur during the first LAG years of each simulation run (i.e. for years before the model can have generated a KKK-LAGth year spawning population except from initial age structure) provision must be made to have EGPROD return these recruitment values rather than ones normally calculated with the stock-recruitment relation. For example, if LAG=3,

```
     SUBROUTINE EGPROD
        .
        .
        .
```

POPSIM

AGE CLASSES AT START OF EACH YEAR

YR	N1(0)	N2(0)	N3(0)		N9(0)	N10(0)	YIELD	TOT N(0)	YOUNG
1	40.000	30.000	25.000		5.000	3.000	50.373	185.000	40.000
2	136.951	28.757	20.962		5.945	2.973	40.163	256.258	136.951
3	136.951	98.458	20.093		4.948	3.535	32.696	309.972	136.951
4	136.951	98.458	68.795	...	3.362	2.942	31.375	348.613	136.951
5	136.951	98.458	68.795		2.563	1.999	35.762	375.437	136.951
6	134.687	98.458	68.795		1.788	1.524	42.516	390.964	134.687

Editor's Note: Output for years 7 through 26 and age groups N4 through N8 has been deleted in this example to conserve space.

27	37.302	27.179	18.895		.369	.145	36.881	106.731	37.302
28	36.008	26.817	18.568		.315	.140	33.162	102.477	36.008
29	33.012	25.887	18.321	...	.226	.119	31.141	97.190	33.012
30	28.613	23.733	17.685		.166	.086	29.843	89.418	28.613

Editor's Note: Line printer graph of total population and total yield appearing here has been deleted.

P(I) 0. .61765

TOTAL YIELD 1443.57275

GRADIENT 5.9025E+03 1.0212E+02

NORM 5.90339E+03

P CHANGES .14998 .00259

****

YIELD TRIAL 3

P(I) .14998 .62025

TOTAL YIELD 1826.39897

GRADIENT -3.8633E+01 5.1378E+00

NORM 3.89733E+01

P CHANGES -.14869 .01977

****

YIELD TRIAL 4

. . .
. . .
. . .

SENSITIVITY BELOW EVALUATED AT BEST PS= .14918 .62672

SENSITIVITY OF BEST YIELD TO PARAMETERS

Q(I)	-7.53013	-8.06870	-7.64886	-6.55162	-5.21072	-3.93362	-2.87343	-1.9628
F(I)	.00000	.00000	.00000	.00000	.68404	1.28301	1.65177	1.7547

W	18.26438	BK	25.39204	AR1	28.12795
		S	-2.81605	AR2	-13.44786

POPSIM

Special note

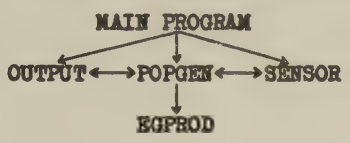
This program contains the statements READ N, list and PRINT N, list which read and write on the system standard input and output units.

Reference

Walters, Carl J., A generalized computer simulation model for fish population studies. 1969 Trans.Amer.Fish.Soc., 98(3):505-512.

PROGRAM LISTING

The program actually consists of a series of subroutines, interrelated as follows:



Function of these subroutines is:

(1) MAIN PROGRAM: controls input, optimizing search procedure, and basic output

(2) OUTPUT: controls output of population structure on every IOCth trial and plots population size and yield

(3) POPGEN: controls changes in population structure during each trial; calculates yield; backstores spawning age structures

(4) SENSOR: performs sensitivity analysis at end of run, if desired

(5) EGPROD: calculates number of young each year, as described above

A listing of the program is given on the following pages.

POPSIM

```
C  FISH POPULATION SIMULATOR--CARL WALTERS, APRIL, 1969
      COMMON/S1/IOG,IMAX,M,K,L,N,KKK,IMIN,IPER,ITL,ITH,W,BK,AR1,AR2,S,WS
     1ENS,BSENS,SSENS,ARSEN1,ARSEN2,YSENS,LAG,F(10),PS(10),Q(10),D1(10),
     2D2(10),D3(10),TAU(10),QW(10,10),BW(10,10),PW(10,10),OY(100),OUT(10
     30,10),PP(100),XF(10),QS(10),QF(10),P(10),Y(10),DUM(10),WO(10),EGS(
     4100),XI(10),OP(100),XS(10),SP(10,10),EGGS
      DIMENSION C(100),PMAX(100),G(100),TITL(10)
 1001 READ 107,(TITL(I),I=1,10)
      PRINT 108,(TITL(I),I=1,10)
      READ 101,IOV,IOG,IOS,IMAX,IOC,M,K,L,IMIN,LAG,IPER,ITL,ITH
      N=K*L
      READ 110,W,BK,AR1,AR2, S,PINC,DELTA,PEWINC,DEL,GMIN,DELMIN
      READ 100,(F(I),I=1,10)
      READ 100,(XI(I),I=1,M)
      READ 100,(PP(I),I=1,K)
      READ 100,(PS(I),I=1,10)
      IF(IOG)990,991,990
  991 READ 100,(Q(I),I=1,M)
      DO 762 I=1,M
      C(I)=.001
      T=I-1+LAG
      D1(I)=EXP(-BK*T)
      D2(I)=EXP(-2.*BK*T)
  762 D3(I)=EXP(-3.*BK*T)
      PRINT 111
      GO  TO 992
  990 READ 100,(TAU(I),I=1,IPER)
      READ 100,((QW(J,I),J=1,10),I=1,M)
      READ 100,((BW(J,I),J=1,10),I=1,M)
      DO 989 I=1,M
      DO 989 J=1,IPER
  989 PW(J,I)=0.
      PRINT 112
  992 DO 883 I=1,N
      OY(I)=0.
      DO 883 J=1,10
  883 OUT(I,J)=0.
      IPET=0
      ITER=0
      IOPT=1
      PEW=1.
      IF(GMIN.EQ.0.) GMIN=.01
      IF(DEL.EQ.0.)DEL=1.0E-5
      IF(L.EQ.0)L=10
      IF(DELMIN.EQ.0.)DELMIN=.001
      IF(DELTA.EQ.0.)DELTA=.3
      DELM=DELTA
      YMAX=0
      PMAX=0
      YTOT=0.
      IF(IOV.EQ.0)PRINT 113
      IF(IOV.GT.0)PRINT 114
      PRINT 102,M,K,L,IMIN
      IF(IOG)150,151,150
  151 PRINT 115,W,BK,(Q(I),I=1,M)
      GO TO 152
  150 PRINT 116,ITL,ITH,(TAU(I),I=1,IPER)
      PRINT 117
      DO 153 I=1,IPER
  153 PRINT 118,I,(QW(I,J),J=1,M)
      PRINT 119
      DO 154 I=1,IPER
  154 PRINT 118,I,(BW(I,J),J=1,M)
```

```
 152 PRINT 120,LAG,S,AR1,AR2,(F(I),I=1,10)
1000 ITER=ITER+1
     IOPT=1
     IF(IPET.EQ.IOC)IPET=0
     IPET=IPET+1
     DO 1 I=1,M
   1 XF(I)=XI(I)
     YLAST=YTOT
     IF(ITER.EQ.1)GOTO 52
     IF(IOV.GT.0)GO TO 52
     DO 53 I=1,K
  53 PP(I)=PP(I)+PINC
  52 PRINT 300,ITER,(PP(I),I=1,K)
     CALL POPGEN(IOPT,YTOT)
     YTEMP=YTOT
     PRINT 501,YTOT
     IF(YTOT.GT.YMAX)998,999
 998 YMAX=YTOT
     DO 993 I=1,K
 993 PMAX(I)=PP(I)
 999 IF(IOV.EQ.0)GO TO 50
     DO 2 I=1,K
     DO 3 J=1,M
   3 XF(J)=XI(J)
     PP(I)=PP(I)+DEL
     CALL POPGEN(0,YTOT)
     G(I)=(YTOT-YTEMP)/DEL
     PP(I)=PP(I)-DEL
   2 IF(PP(I).EQ.0..AND.G(I).LT.0.)G(I)=0.
     PRINT 301,(G(I),I=1,K)
     IF(PEWINC) 35,34,35
  34 IF(YTOT.LT.YLAST) DELM=DELM/2.
     PEW=1.
     GO TO 36
  35 IF(YTOT.LT.YLAST)PEW=PEW+PEWINC
  36 DELTA=DELM/PEW
     GRAD=0.
     DO 4 I=1,K
   4 GRAD=GRAD+G(I)*G(I)
     GRAD=SQRT(GRAD)
     DO 5 I=1,K
     C(I)=DELTA*G(I)/GRAD
     PP(I)=PP(I)+C(I)
     IF(PP(I).LT.0.)PP(I)=0.
   5 CONTINUE
     PRINT 400,GRAD,(C(I),I=1,K)
  50 CONTINUE
     IF(IPET-1)12,11,12
  11 CALL OUTPUT
  12 CONTINUE
     IF(IOV.EQ.0..AND.ITER.EQ.IMAX)GO TO 503
     IF(IOV.EQ.0)GO TO 1000
     ICUM=0
     ICU=0
     DO 8 I=1,K
     IF((ABS(G(I))).LE.GMIN)ICUM=ICUM+1
     IF(ABS(C(I)).LT.DELMIN)ICU=ICU+1
   8 CONTINUE
     IF(ICU.LT.K.AND.ICUM.LT.K) GO TO 31
     IF(IPET.GT.1)CALL OUTPUT
     PRINT 202
     GO TO 503
```

```
 31 IF(ITER.LE.IMAX) GO TO 1000
    IF(IPET.GT.1)CALL OUTPUT
    PRINT 203
503 IF(IOS.GT.0.AND.IOG.EQ.0)522,1001
522 YSENS=YMAX
    DO 560 I=1,K
560 PP(I)=PMAX(I)
    PRINT 513,(PMAX(I),I=1,K)
    CALL SENSOR
    PRINT 510,(QS(I),I=1,M)
    PRINT 511,(QF(I),I=1,10)
    PRINT 512,WSENS,BSENS,SSENS,ARSEN1,ARSEN2
505 GO TO 1001
100 FORMAT(10F5.5)
101 FORMAT(3I1,I3,9I2)
102 FORMAT(1HA,50X,13HNO AGE GROUPS,I17/1HA,50X,16HNO PERIODS/TRIAL,I1
   13/1HA,50X,13HPERIOD LENGTH,I17/1HA,50X,22HFIRST AGE GROUP FISHED,I
   28)
107 FORMAT(10A8)
108 FORMAT(1H1,25X,10A8)
110 FORMAT(6F5.5,5E10.5)
111 FORMAT(1H0,50X,41HGROWTH-MORTALITY TYPE  BERTALANFFY-ANNUAL)
112 FORMAT(1H0,50X,32HGROWTH-MORTALITY TYPE  SEGMENTAL)
113 FORMAT(1HA,50X,32HSEARCH PROCEDURE       SYSTEMATIC)
114 FORMAT(1HA,50X,28HSEARCH PROCEDURE       ASCENT)
115 FORMAT(38H0GROWTH PARAMETERS   ASYMPTOTIC WEIGHT,F10.5/1HA,20X,8HG
   1ROWTH K,F19.5/24H0NATURAL MORTALITY RATES,10F10.5)
116 FORMAT(1HA,50X,27HPERIODS FISHED IN EACH YEAR,,I3,7HTHROUGH,I3/32H
   10GROWTH-MORTALITY PERIOD LENGTHS,10F10.5)
117 FORMAT(31H0NATURAL MORTALITY RATES BY AGE)
118 FORMAT(1HA,2X,7HPERIOD(,I2,1H),10F10.5)
119 FORMAT(20H0GROWTH RATES BY AGE)
120 FORMAT(24H0REPRODUCTIVE PARAMETERS/1HA,5X,26HTIME LAG,BIRTH-RECRUI
   1TMENT,I5/1HA,5X,13HSPAWNING TIME,F10.5/1HA,5X,3HAR1,F20.5/1HA,5X,3
   2HAR2,F20.5/1HA,5X,4HF(I),10F10.5)
202 FORMAT(42H0THIS SOLUTION MEETS CRITERION FOR OPTIMUM)
203 FORMAT(38H0MAXIMUM NUMBER OF ITERATIONS EXCEEDED)
300 FORMAT(1H0,100(1H*)/12H0YIELD TRIAL,I5/5H0P(I),10F10.5)
301 FORMAT(9H0GRADIENT,10E12.4)
400 FORMAT(5H0NORM,E20.5/10H0P CHANGES,10F10.5)
501 FORMAT(12H0TOTAL YIELD,F20.5)
510 FORMAT(1H0,20X,39HSENSITIVITY OF BEST YIELD TO PARAMETERS/5H0Q(I),
   110F13.5)
511 FORMAT(5H0F(I),10F13.5)
512 FORMAT(2H0W,F15.5/3H0BK,F15.5/2H0S,F15.5/4H0AR1,F15.5/4H0AR2,F15.5
   1)
513 FORMAT(1H1,130(1H=)/40H0SENSITIVITY BELOW EVALUATED AT BEST PS=,10
   1F12.5/1HA,12F10.5/1HA,12F10.5/1HA,12F10.5/1HA,12F10.5/1HA,12F10.5/
   21HA,12F10.5/1HA,12F10.5/1HA,12F10.5)
    END


    SUBROUTINE POPGEN(IOPT,YTOT)
    COMMON/S1/IOG,IMAX,M,K,L,N,KKK,IMIN,IPER,ITL,ITH,W,BK,AR1,AR2,S,WS
   1ENS,BSENS,SSENS,ARSEN1,ARSEN2,YSENS,LAG,F(10),PS(10),Q(10),D1(10),
   2D2(10),D3(10),TAU(10),QW(10,10),BW(10,10),PW(10,10),OY(100),OUT(10
   30,10),PP(100),XF(10),QS(10),QF(10),P(10),Y(10),DUM(10),WO(10),EGS(
   4100),XI(10),OP(100),XS(10),SP(10,10),EGGS
    DIMENSION          FA(10),FB(10),YA(10),YB(10),YC(10),DUW(10,10),DU
   1S(10),ISP(10),        XNT(10)
    YTOT=0
    KKK=0
```

```
      DO 100 KPER=1,K
      IF(IOG.GT.0)GO TO 40
      DO 20 I=1,M
      P(I)=0.
      IF(I.GE.IMIN)P(I)=PP(KPER)*PS(I-IMIN+1)
      FA(I)=EXP(-(Q(I)+P(I))*S)
      FB(I)=EXP(-Q(I)-P(I))
      YA(I)=EXP(-P(I)-Q(I)-BK)
      YB(I)=EXP(-P(I)-Q(I)-2.*BK)
   20 YC(I)=EXP(-P(I)-Q(I)-3.*BK)
      DO 101 LPER=1,L
      KKK=KKK+1
      YT=0
      POP=0
      DO 1 I=1,M
      POP=POP+XF(I)
      XS(I)=XF(I)*FA(I)
      Y(I)=0.
      IF(I.LT.IMIN)GO TO 1
      Z=P(I)+Q(I)
      AA=(1.-FB(I))/Z
      AB=-3.*D1(I)*(1.-YA(I))/(Z+BK)
      AC=3.*D2(I)*(1.-YB(I))/(Z+2.*BK)
      AD=-D3(I)*(1.-YC(I))/(Z+3.*BK)
      Y(I)=P(I)*XF(I)*W*(AA+AB+AC+AD)
    1 YT=YT+Y(I)
      IF(KKK.GT.1.OR.LAG.EQ.0)GO TO 111
      DO 112 I=1,LAG
      DO 112 J=1,M
  112 SP(I,J)=XS(J)
  111 CONTINUE
      CALL EGPROD
      IF(LAG.EQ.0)GO TO 114
      LLL=LAG+1
      DO 113 I=1,LAG
      DO 113 J=1,M
      IF(I.LT.LAG)SP(LLL-I,J)=SP(LLL-I-1,J)
      IF(I.EQ.LAG)SP(1,J)=XS(J)
  113 CONTINUE
  114 CONTINUE
      IF(IOPT)3,3,2
    2 DO 33 KK=1,M
   33 OUT(KKK,KK)=XF(KK)
      EGS(KKK)=XF(1)
      OP(KKK)=POP
      OY(KKK)=YT
    3 YTOT=YTOT+YT
   10 DO 12I=1,M
   12 DUM(I)=XF(I)
      DO 13 I=2,M
   13 XF(I)=DUM(I-1)*FB(I-1)
      XF(1)=EGGS
  101 CONTINUE
      GO TO 100
   40 DO 53 J=1,M
      TTOT=0
      DO 53 I=1,IPER
      IF(J.LT.IMIN)GO TO 54
      IF(I.GE.ITL.AND.1.LE.ITH)PW(I,J)=PP(KPER)*PS(J-IMIN+1)
   54 IF(TTOT.LE.S.AND.(TTOT+TAU(I)).GT.S)44,45
   44 DUS(J)=EXP(-(PW(I,J)+QW(I,J))*(S-TTOT))
      ISP(J)=I
```

```
   45 TTOT=TTOT+TAU(I)
   53 DUW(I,J)=EXP(-(PW(I,J)+QW(I,J))*TAU(I))
      DO 102 LYR=1,L
      KKK=KKK+1
      WO(1)=BK
      YT=0
      POP=0
      DO 41 IYR=1,M
      XNT(1)=XF(IYR)
      OUT(KKK,IYR)=XF(IYR)
      POP=POP+XF(IYR)
      DO 42 I=2,IPER
      XNT(I)=XNT(I-1)*DUW(I-1,IYR)
   42 WO(I)=WO(I-1)+BW(I-1,IYR)*TAU(I-1)
      ITEMP=ISP(IYR)
      XS(IYR)=XNT(ITEMP)*DUS(IYR)
      YAGE=0
      IF(IYR.LT.IMIN)GO TO 60
      DO 43 I=1,IPER
      Z=PW(I,IYR)+QW(I,IYR)
   43 YAGE=YAGE+(PW(I,IYR)/Z)*XNT(I)*((WO(I)+BW(I,IYR)/Z)-DUW(I,IYR)
     1*(WO(I)+BW(I,IYR)*TAU(I)+BW(I,IYR)/Z))
   60 Y(IYR)=YAGE
      YT=YT+Y(IYR)
      WO(1)=WO(IPER)+BW(IPER,IYR)*TAU(IPER)
      IF(IYR.LT.M)DUM(IYR+1)=XNT(IPER)*DUW(IPER,IYR)
   41 CONTINUE
      OP(KKK)=POP
      IF(KKK.GT.1.OR.LAG.EQ.0)GO TO 115
      DO 116 I=1,LAG
      DO 116 J=1,M
  116 SP(I,J)=XS(J)
  115 CONTINUE
      CALL EGPROD
      IF(LAG.EQ.0)GO TO 117
      LLL=LAG+1
      DO 118 I=1,LAG
      DO 118 J=1,M
      IF(I.LT.LAG)SP(LLL-I,J)=SP(LLL-I-1,J)
      IF(I.EQ.LAG)SP(1,J)=XS(J)
  118 CONTINUE
  117 CONTINUE
      XF(1)=EGGS
      DO 47 I=2,M
   47 XF(I)=DUM(I)
      YTOT=YTOT+YT
      EGS(KKK)=EGGS
      OY(KKK)=YT
  102 CONTINUE
  100 CONTINUE
      RETURN
      END


      SUBROUTINE OUTPUT
      DIMENSION ALINE (101)
      COMMON/S1/IOG,IMAX,M,K,L,N,KKK,IMIN,IPER,ITL,ITH,W,BK,AR1,AR2,S,WS
     1ENS,BSENS,SSENS,ARSEN1,ARSEN2,YSENS,LAG,F(10),PS(10),Q(10),D1(10),
     2D2(10),D3(10),TAU(10),QW(10,10),BW(10,10),PW(10,10),OY(100),OUT(10
     30,10),PP(100),XF(10),QS(10),QF(10),P(10),Y(10),DUM(10),WO(10),EGS(
     4100),XI(10),OP(100),XS(10),SP(10,10),EGGS
      DATA YAXIS,XAXIS,FILL,CN,CY,CR/1H-,1HI,1H ,1H*,1H.,1H0/
      PRINT 400
```

POP31N

```
400 FORMAT(1H0,20X,33HAGE CLASSES AT START OF EACH YEAR/3H0YR,5X,5HN1(
   10),5X,5HN2(0),5X,5HN3(0),5X,5HN4(0),5X,5HN5(0),5X,5HN6(0),5X,5HN7(
   20),5X,5HN8(0),5X,5HN9(0),4X,6HN10(0),5X,25HYIELD  TOT N(0)      YOU
   3NG)
    YMN=0.
    YMY=0.
 17 DO 34 I=1,N
    IF(OP(I)-YMN )55,55,66
 66 YMN=OP(I)
 55 IF(OY(I)-YMY )77,77,88
 88 YMY=OY(I)
 77 CONTINUE
 34 PRINT 100,I,(OUT(I,J),J=1,10),OY(I),OP(I),EGS(I)
100 FORMAT (1HA,I2,13F10.3)
    PRINT 104,YMN,YMY
104 FORMAT(1H1,34X,26HTOTAL POPULATION (*), MAX=,F20.5/5HAYEAR,30X,21H
   1TOTAL YIELD (.), MAX=,F20.5)
    IF(YMN)53,54,53
 54 YMN=1.
 53 IF(YMY)56,57,56
 57 YMY=1.
 56 CONTINUE
    KK=1
    DO 4J=1,101
    IF(J-10*KK-1)20,21,20
 21 KK=KK+1
    ALINE(J)=XAXIS
    GO TO 4
 20 ALINE(J)=YAXIS
  4 CONTINUE
    PRINT 101,(ALINE(I),I=1,101)
101 FORMAT(1HA,4X,101A1)
    DO 6J=1,101
  6 ALINE(J)=FILL
    ALINE(1)=XAXIS
    DO 8LI=1,N
    J=100.*(OP(LI)/YMN)+1.5
    JJ=100.*(OY(LI)/YMY)+1.5
    IF(J-JJ)1,2,1
  2 ALINE(J)=CR
    GO TO 3
  1 ALINE(J)=CN
    ALINE(JJ)=CY
  3 PRINT 102,LI,(ALINE(KK),KK=1,101)
102 FORMAT(1HA,I3,1X,101A1)
    ALINE(J)=FILL
    ALINE(JJ)=FILL
    ALINE(1)=XAXIS
  8 CONTINUE
    DO 11J=1,101
 11 ALINE(J)=YAXIS
    PRINT 101,(ALINE(I),I=1,101)
    RETURN
    END


    SUBROUTINE SENSOR
    COMMON/S1/IOG,IMAX,M,K,L,N,KKK,IMIN,IPER,ITL,ITH,W,BK,AR1,AR2,S,WS
   1ENS,BSENS,SSENS,ARSEN1,ARSEN2,YSENS,LAG,F(10),PS(10),Q(10),D1(10),
   2D2(10),D3(10),TAU(10),QW(10,10),BW(10,10),PW(10,10),OY(100),OUT(10
   30,10),PP(100),XF(10),QS(10),QF(10),P(10),Y(10),DUM(10),WO(10),EGS(
   4100),XI(10),OP(100),XS(10),SP(10,10),EGGS
    DO 1I=1,10
    IF(I.GT.M)GO TO 15
    DQ=.01*Q(I)
    Q(I)=Q(I)+DQ
    DO 5 J=1,M
```

```
    5 XF(J)=XI(J)
      CALL POPGEN(0,YTOT)
      QS(I)=(YTOT-YSENS)
      Q(I)=Q(I)-DQ
   15 CONTINUE
      FQ=.01*F(I)
      F(I)=F(I)+FQ
      DO 6 J=1,M
    6 XF(J)=XI(J)
      CALL POPGEN(0,YTOT)
      QF(I)=(YTOT-YSENS)
    1 F(I)=F(I)-FQ
      WQ=.01*W
      W=W+WQ
      DO 8 J=1,M
    8 XF(J)=XI(J)
      CALL POPGEN(0,YTOT)
      WSENS=(YTOT-YSENS)
      W=W-WQ
      SQ=.01*S
      DO 9 J=1,M
    9 XF(J)=XI(J)
      IF(S-1.)2,3,3
    2 S=S+SQ
      CALL POPGEN(0,YTOT)
      SSENS=(YTOT-YSENS)
      S=S-SQ
      GO TO 4
    3 S=S-SQ
      CALL POPGEN(0,YTOT)
      SSENS=(YSENS-YTOT)
      S=S+SQ
    4 AQ=.01*AR1
      DO 10 J=1,M
   10 XF(J)=XI(J)
      AR1=AR1+AQ
      CALL POPGEN(0,YTOT)
      ARSEN1=YTOT-YSENS
      AR1=AR1-AQ
      AQ=.01*AR2
      DO 11 J=1,M
   11 XF(J)=XI(J)
      AR2=AR2+AQ
      CALL POPGEN(0,YTOT)
      ARSEN2=YTOT-YSENS
      AR2=AR2-AQ
      BQ=.01*BK
      BK=BK+BQ
      DO 7 I=1,M
      XF(I)=XI(I)
      T=I-1+LAG
      D1(I)=EXP(-BK*T)
      D2(I)=EXP(-2.*BK*T)
    7 D3(I)=EXP(-3.*BK*T)
      CALL POPGEN(0,YTOT)
      BSENS=(YTOT-YSENS)
      BK=BK-BQ
      RETURN
      END
```

```
      SUBROUTINE EGPROD
      COMMON/S1/IOG,IMAX,M,K,L,N,KKK,IMIN,IPER,ITL,ITH,W,BK,AR1,AR2,S,WS
     1ENS,BSENS,SSENS,ARSEN1,ARSEN2,YSENS,LAG,F(10),PS(10),Q(10),D1(10),
     2D2(10),D3(10),TAU(10),QW(10,10),BW(10,10),PW(10,10),OY(100),OUT(10
     30,10),PP(100),XF(10),QS(10),QF(10),P(10),Y(10),DUM(10),WO(10),EGS(
     4100),XI(10),OP(100),XS(10),SP(10,10),EGGS
      E=0.
      DO 1 I=5,10
    1 E=E+SP(LAG,I)*F(I)
      EGGS=E*EXP(AR1*(1.-E **AR2))
      RETURN
      END
```

Program name - RICKER 1

Purpose

Fits the Ricker spawner-recruit curve and prints a table of fitted values.

Programmer - Patrick K. Tomlinson

Originating organization - California Department of Fish and Game

Approximate storage - 1,000 (CDC 1604)

This program estimates the parameters of the Ricker spawner-recruit curve, $R = ASe^{-bS}$, from fitting the logarithmic transformation, $\ln (R/S) = \ln A - bS$, by the method of least squares. S is the spawning biomass, R is the recruit biomass, and A and b are constants. From the fitted curve a table of spawning stocks and resultant recruitments is produced. The curve is discussed in Ricker (1958).

The cards used with the program are defined in terms of sets:

Set #1	1 card	FORMAT (10A8)
	Col. 1	Numeral one (carriage control).
	Cols. 2-80	Alphabetic title information such as type of curve and species.
Set #2	1 card	FORMAT (2I4, 2F10.0)
	Cols. 1-4	Number (N) of observed pairs (R, S). Right-justify with no decimal.
	Cols. 5-8	Number (N) of estimates of R desired in output table. Right-justify with no decimal.
	Cols. 9-18	Initial value of S to be used in generating estimates of R in the output table. Left-justify with decimal punched.
	Cols. 19-28	Increment (Δ) for S ($S_{i+1} = S_i + \Delta$) to be used in generating all additional R in the output table. Left-justify with decimal punched.
Set #3	$1 \leq$ cards ≤ 13	FORMAT (8F10.0)
	Cols. 1-80	Each card has 8 fields, with each field containing 10 columns. The observed spawning biomasses are punched with S(1) in field 1, S(2) in field 2, ..., S(N) in field N. Punch eight values per card, left-justify with a decimal point.
Set #4	$1 \leq$ cards ≤ 13	FORMAT (8F10.0)
	Cols. 1-80	Like Set #3, only each field contains the observed recruits corresponding to the spawners in Set #3.

** Repeat all cards for additional jobs.
No error messages.

Special notes

This program reads and writes on the standard input and output units with the statements READ N, list and PRINT N, list. It contains an end-of-file check at statement 7+1 but should operate satisfactorily if the statement is removed. Also contained are $ statement separators and a multiple replacement statement. Statement functions:

EXPF(X) = EXP(X) and

LOGF(X) = ALOG(X)

should be inserted after the dimension statement for FORTRAN IV operation.

RICKER 1

Reference

Ricker, W.E., Handbook of computations for biological statistics of fish populations.
1958 Bull.Fish.Res.Bd Canada, (119):1-300.

```
      PROGRAM RICKER1 (INPUT,OUTPUT,TAPE50=INPUT)
      DIMENSION X(100),Y(100),Z(100),NAME(10)
   11 FORMAT (10A8)
    7 READ 11,(NAME(I),I=1,10) $ IF(EOF,50) 13,14
   10 FORMAT (2I4,2F10.0)
   14 READ 10,N,M,D,H $ FN=N
      SX=SY=SXY=SXS=0.
      READ 2,(X(I),I=1,N)
      READ 2,(Z(I),I=1,N)
    2 FORMAT (8F10.0)
      DO 3 I=1,N  $ Y(I)=LOGF(Z(I)/X(I)) $ SY=SY+Y(I) $SXY=SXY+X(I)*Y(I)
      SX=SX+X(I)
    3 SXS=SXS+X(I)**2
      B=(SX*SY-FN *SXY)/(FN *SXS-SX**2)
      A=(SY+B*SX)/FN
      C=EXPF(A)
      PRINT 11,(NAME(I),I=1,10)
      PRINT 6,C,B
      PRINT 5
      DO 4 I=1,M
      E=C*D*EXPF(-B*D)
      PRINT 8,D,E
    4 D=D+H
      GOTO 7
    8 FORMAT (1H0,2(10X,F8.4))
    5 FORMAT(1H0,80(1H*)/1H0,10X8HSPAWNERS,10X8HRECRUITS/1H0,80(1H*))
    6 FORMAT (1H0,80(1H*)/1H0,9HRECRUITS=F7.4,15H(SPAWNERS)EXP(-,F7.5,11
     1H(SPAWNERS)))
   13 STOP
      END
```

Program name – WTLN

Purpose

Fits a curve giving weight as a function of length, produces a table of fitted weights and lengths and provides various related statistics.

Programmer – Norman J. Abramson
Modified by Patrick K. Tomlinson and Catherine L. Berude

Originating organization – California Department of Fish and Game

Approximate storage – 16,500 (CDC 6600)

The main program, WTLN, controls data input and output, calls subroutines WTLGTH and MINMAX, and references function T05. Data decks may be stacked for repeated execution without program recompilation.

Subroutine WTLGTH fits a curve of the form $W = a L^b$, where W = weight and L = length to data containing from 4 to 5000 pairs of observed length-weight. The method of fitting involves linearization by common logarithms and the usual least squares procedure for fitting a straight line.

Subroutine MINMAX determines the range of the observed data by searching for the shortest, longest, lightest, and heaviest fish. Those four observations are flagged and the pairs of length-weight for each are output.

Function T05 computes the Student's t value at probability level 0.05 (two-sided) for N-2 degrees of freedom.

Output:
Printed output includes estimates of the exponent and coefficient of the fitted curve, various test values and intermediate values, minimum and maximum observed lengths and weights, and a table of 200 weights computed from the fitted curve for a variable range of lengths. Output values are well-labelled. Lenths are printed with one decimal place, and weights with two decimal places.

Input:
Input to the program consists of program control cards and cards containing observed length and weight measurement data. The observed data may be in frequency format or in pairs of (length, weight); the format specification by which the data are read may be either the fixed format written into the program or a variable specification input on a program control card.

a. TITLE — REQUIRED
one card FORMAT (10A8)

Columns 1-80	Alphanumeric description of job (units of measure, species, collection period, etc.) which will be printed as a title. Center data on column 40.

b. REQUIRED
one card FORMAT (2A5, 2F10.0)

Columns 1-3	Option determining use of fixed or variable format. Punch FIX if the built-in format (2F10.0), applies; or, punch VAR if the data input specification is supplied by the user.
Columns 4-5	Blank
Columns 6-9	Data form indicator. Enter PAIR if data are punched as pairs of length-weight; or, specify FREQ if data are punched in frequency format (length, frequency, weights).
Column 10	Blank
Columns 11-20	Left-justified with decimal punched, the lowest length to be used in computing 200 estimated weights from the estimated equation.

Columns 21-30 Left-justified with decimal punched, the increment between lengths for computing the expected weights. This increment is added 199 times, beginning with the first addition to the lowest length value punched in columns 11-20 of this card.

c. *OPTIONAL (included only if VAR is punched in columns 1-3 of card b.)
one card FORMAT (I2, 6X, 9A8)

Columns 1-2 Right-justified without decimal, the number of pairs of (length, weight) punched on each data card. If data are in frequency format, the value in this field will be ignored.
Columns 3-8 Blank
Columns 9-80 The format according to which the observed data will be read. The first punch must be a left parenthesis, and the final punch a right parenthesis.

d. REQUIRED
1-5000 cards FORMAT: either (2F10.0), or that specified on card c.

Observed data. Pairs must be punched as (length, weight). Data must be punched as: length (floating-point), frequency (integer), and observed weights (floating-point) where "frequency" is the number of weights listed at that length; the format specification of card c must take into account possible continuation card(s) necessitated by the number of weights observed at a length.

e. REQUIRED
one card FORMAT: same as deck d.

Columns 1-80 Blanks, to signal the end of observation data.

Units a - e are repeated for each additional set of observed data to be analyzed. Following the last such set, and end-of-job indicator must be present:

f. REQUIRED
one card FORMAT (10A8). This card will be read as a potential card a.

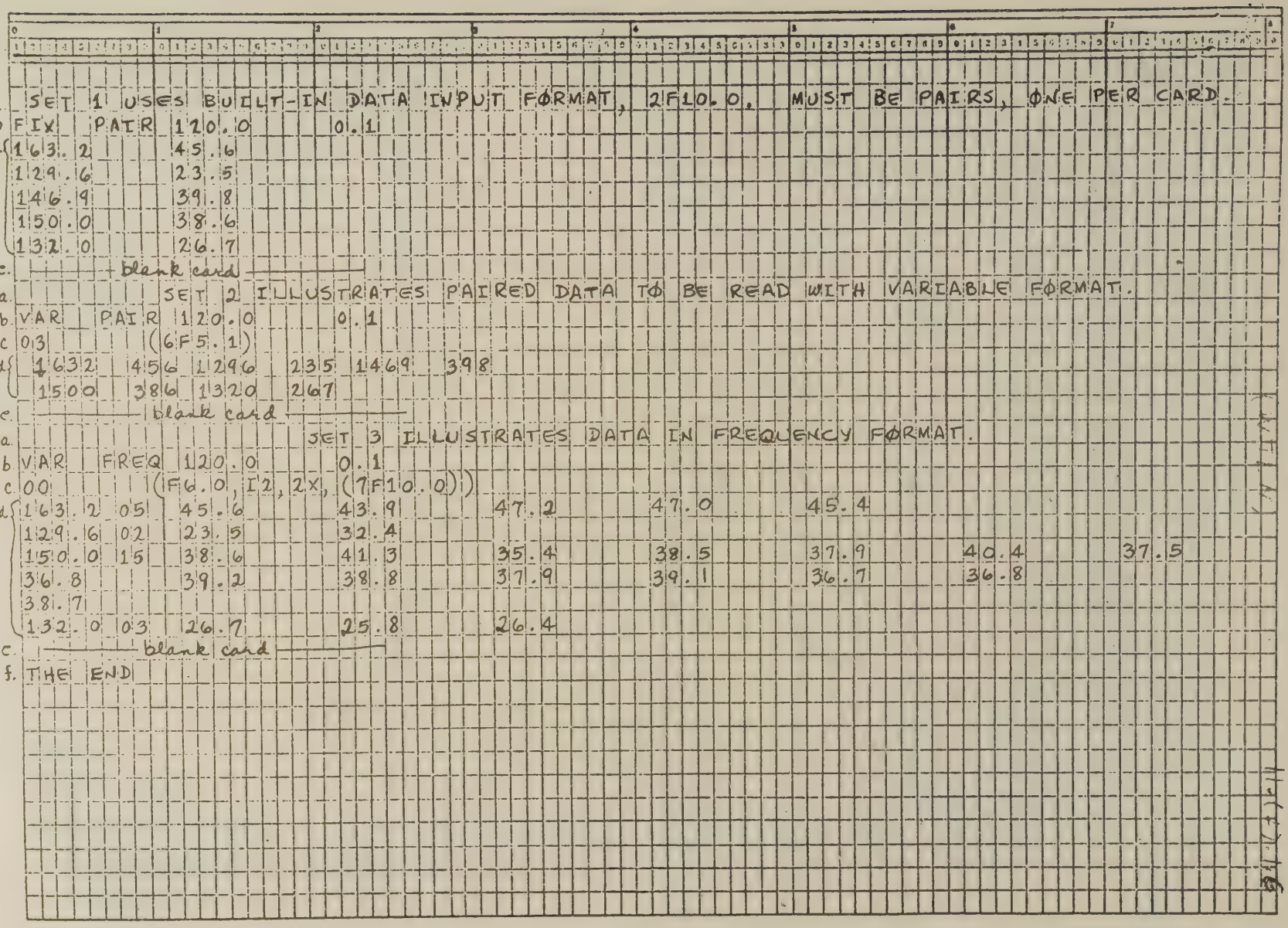
Columns 1-7,
THE END

* If columns 1-3 of card b contain FIX, card c must be omitted. Card c is included only if VAR is punched in columns 1-3 of card b.

The attached describes the input deck set-up for a single job containing three separate sets of data. Each set illustrated the deck set-up for a different data format configuration.

Special notes

Storage reduction may be accomplished by lowering the dimensions of S and W in the main program and subroutines WTLGTH and MINMAX to the maximum number of observations required. The statements READ N, list and PRINT N, list appear in this program.


1a SET 1 USES BUILT-IN DATA INPUT FORMAT, 2F10.0, MUST BE PAIRS, ONE PER CARD.
b FIX PAIR 120.0 0.1
d 163.2 45.6
129.6 23.5
146.9 39.8
150.0 38.6
132.0 26.7
c. blank card
2a. SET 2 ILLUSTRATES PAIRED DATA TO BE READ WITH VARIABLE FORMAT.
b VAR PAIR 120.0 0.1
c 03 (6F5.1)
d 1632 456 1296 235 1469 398
1500 386 1320 267
e. blank card
3a. SET 3 ILLUSTRATES DATA IN FREQUENCY FORMAT.
b VAR FREQ 120.0 0.1
c 00 (F6.0, I2, 2X, (7F10.0))
d 163.2 05 45.6 43.9 47.2 47.0 45.4
129.6 02 23.5 32.4
150.0 15 38.6 41.3 35.4 38.5 37.9 40.4 37.5
36.8 39.2 38.8 37.9 39.1 36.7 36.8
38.7
132.0 03 26.7 25.8 26.4
c. blank card
f. THE END

11.(2).1.3

WTLN

```
      PROGRAM WTLN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C WEIGHT-LENGTH REGRESSION
C
C     EXPECTED WEIGHTS ARE CALCULATED FOR 200 LENGTHS (RANGE VARIABLE),
C     MINIMUM AND MAXIMUM OBSERVED WEIGHT AND LENGTH DETERMINED.
C
      COMMON S,W
      DIMENSION FMAT(9),TITLE(10),S(5000),W(5000),FLGTH(200),WT(200)
      DIMENSION  Q(30)
      INTEGER FREQ
    5 CONTINUE
  700 READ 800, (TITLE(I),I=1,10)
  800 FORMAT (10A8)
      IF(TITLE(1).EQ.7HTHE END) GO TO 503
      READ 801, FMTOPT,LWFORM,FLGTH(1),FLINC
  801 FORMAT (2A5,2F10.0)
      IF(FMTOPT.EQ.3HVAR) GO TO 705
      FMAT(1)= 8H(2F10.0)
      DO 701 I=2,9
      FMAT(I)= 1H
  701 CONTINUE
      NPC=1
      LWFORM=4HPAIR
      GO TO 710
  705 READ 802, NPC,(FMAT(I),I=1,9)
  802 FORMAT (I2,6X,9A8)
  710 IF(LWFORM.EQ.4HFREQ) GO TO 715
      DO 714 I=1,5000,NPC
      LU=I+NPC-1
      READ FMAT, (S(J),W(J),J=I,LU)
      IF(S(I)) 711,711,714
  711 K=I-NPC+1
      DO 713 J=K,I
      IF(S(J)) 712,712,713
  712 N=J
      GO TO 720
  713 CONTINUE
  714 CONTINUE
      PRINT 850
  850 FORMAT (1H1,8(/),20X,65H*****  NUMBER OF PAIRS OF OBSERVED DATA GR
     1EATER THAN 5000  *****  )
  715 N=1
      DO 718 I=1,5000
      READ FMAT, FL,FREQ,(Q(J),J=1,FREQ)
      IF(FL) 720,720,716
  716 K=0
      LU=N+FREQ-1
      DO 717 J=N,LU
      K=K+1
      S(J)=FL
      W(J)=Q(K)
  717 CONTINUE
      N=N+FREQ
  718 CONTINUE
  720 N=N-1
  725 NDF = N-2
      T05=T05(NDF)
```

```
207 CALL WTLGTH (N,T05,AL,A,B,SE,SEB,CL,CU,T)
    CALL MINMAX (N,FMINLL,FMINLW,FMAXLL,FMAXLW,FMINWL,FMINWW,
   1 FMAXWL,FMAXWW)
    PRINT 7, (TITLE(I),I=1,10),A,FMINLL,FMINLW,AL,FMAXLL,FMAXLW,B,
   1FMINWL,FMINWW,SEB,FMAXWL,FMAXWW,CU,CL,N,T05,SE,T
  7 FORMAT (1H1,42X,33HLENGTH-WEIGHT ANALYSIS   W=A*L**B/20X,10A8//
   A1X,120(1H*)//93X,17HLENGTH --- WEIGHT/10X,32HCOEFFICIENT A OF FITT
   BED CURVE = ,E13.5,10X,23HMINIMUM LENGTH OBSERVED,2F11.3/34X,8HLOG
   CA = ,F13.5,10X,23HMAXIMUM LENGTH OBSERVED,2F11.3//10X,29HEXPONENT
   DB OF FITTED CURVE = ,F16.5,10X,23HMINIMUM WEIGHT OBSERVED,2F11.3/
   E17X,22HSTANDARD ERROR OF B = ,F16.5,10X,23HMAXIMUM WEIGHT OBSERVED
   F,2F11.3/10X,29H0.95 UPPER CONF LIMIT OF B = ,F16.5/10X,29H0.95 LOW
   GER CONF LIMIT OF B = ,F16.5/65X,27HNUMBER OF PAIRS OBSERVED = ,I18
   H/65X,29HSTUDENTS T (0.95, 2-SIDED) = ,F16.5/10X,33HSTANDARD ERROR
   IOF THE ESTIMATE = ,F12.5,10X,28HCOMPUTED T FOR TEST, B=3  = ,F17.5
   J//1X,120(1H*)//42X,38H200 EXPECTED WEIGHTS FOR GIVEN LENGTHS//16X,
   K8HEXPECTED,23X,8HEXPECTED,23X,8HEXPECTED,23X,8HEXPECTED/4X,6HLENGT
   LH,7X,6HWEIGHT,3(12X,6HLENGTH,7X,6HWEIGHT)/)
    WT(1)=A*FLGTH(1)**B
    DO 51 I=2,200
    FLGTH(I)=FLGTH(I-1)+FLINC
 51 WT(I)=A*FLGTH(I)**B
    PRINT 31, (FLGTH(I),WT(I),FLGTH(I+49),WT(I+49),FLGTH(I+99),
   1WT(I+99),FLGTH(I+149),WT(I+149),I=1,35)
 31 FORMAT(F10.1,F13.2,3(F18.3,F13.2))
    PRINT 32, (FLGTH(I),WT(I),FLGTH(I+49),WT(I+49),FLGTH(I+99),
   AWT(I+99),FLGTH(I+149),WT(I+149),I=36,50)
 32 FORMAT (1H1,15X,8HEXPECTED,3(23X,8HEXPECTED)/4X,6HLENGTH,7X,6HWEIG
   1HT,3(12X,6HLENGTH,7X,6HWEIGHT)//(F10.1,F13.2,F18.1,F13.2 ,F18.1,
   2F13.2,F18.1,F13.2))
    PRINT 33
 33 FORMAT (// 1X,120(1H*))
    GO TO 5
503 STOP
    END



    SUBROUTINE WTLGTH (N,T05,AL,A,B,SE,SEB,CL,CU,T)
    DIMENSION S(5000),W(5000)
    COMMON S,W
    FN=N
    BNUM=0.
    BDENOM=0.
    SEINT=0.
    SUMS=ALOG10(S(1))
    SUMW=ALOG10(W(1))
    DO 5 I=2,N
    SUMS=SUMS+ALOG10(S(I))
    SUMW=SUMW+ALOG10(W(I))
  5 CONTINUE
    SUMS=SUMS/FN
    SUMW=SUMW/FN
    DO 8 I=1,N
    SDIF=ALOG10(S(I))-SUMS
    WDIF=ALOG10(W(I))-SUMW
    BNUM=BNUM+ SDIF*WDIF
    BDENOM=BDENOM+ SDIF**2
    SEINT=SEINT+ WDIF**2
```

```
    8 CONTINUE
      B=BNUM/BDENOM
      AL=SUMW - (B*SUMS)
      A=10.**AL
      VREG=0.0
      DO 17 I=1,N
   17 VREG=VREG+(W(I)-A*S(I)**B)**2
      SE=SQRT(VREG/(FN-2.))
      SESQ=SEINT-(BNUM**2 / BDENOM)
      SESQ=SESQ/((FN-2.)*BDENOM)
      SEB=SQRT(SESQ)
      CL=B-T05*SEB
      CU=B+T05*SEB
      T=(B-3.)/SEB
      RETURN
      END



      SUBROUTINE MINMAX (NS,FINS,FINSW,FAXS,FAXSW,FINW,FINWS,FAXW,FAXWS)
      DIMENSION S(5000),W(5000)
      COMMON S,W
      FINS=S(1)
      FINSW=W(1)
      FAXS=S(1)
      FAXSW=W(1)
      FINW=W(1)
      FINWS=S(1)
      FAXW=W(1)
      FAXWS=S(1)
      DO 400 I=2,NS
      IF(FINS-S(I)) 300,300,301
  301 FINS=S(I)
      FINSW=W(I)
  300 IF(FAXS-S(I)) 302,303,303
  302 FAXS=S(I)
      FAXSW=W(I)
  303 IF(FINW-W(I)) 304,304,305
  305 FINW=W(I)
      FINWS=S(I)
  304 IF(FAXW-W(I)) 306,400,400
  306 FAXW=W(I)
      FAXWS=S(I)
  400 CONTINUE
      RETURN
      END



      FUNCTION T05 (NN)
      DIMENSION  NVAL(45),T(45)
      DATA  NVAL / 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21
     A,22,23,24,25,26,27,28,29,30,35,40,45,50,55,60,70,80,90,100,200,300
     B,400,500,1000/
      DATA  T   /12.706,4.303,3.182,2.776,2.571,2.447,2.365,2.306,2.262,
     A2.228,2.201,2.179,2.160,2.145,2.131,2.120,2.110,2.101,2.093,2.086,
     B2.080,2.074,2.069,2.064,2.060,2.056,2.052,2.048,2.045,2.042,2.030,
     C2.021,2.014,2.008,2.004,2.000,1.994,1.990,1.987,1.984,1.972,1.968,
     D1.966,1.965,1.962/
      IF(NN-30) 5,5,6
```

WTLN

```
    5 TO5=T(NN)
      GO TO 20
    6 IF(NN-1000) 10,10,8
    8 TO5=1.95999
      GO TO 20
   10 DO 15 I=31,45
      IF(NN-NVAL(I)) 12,13,15
   12 XLO=NVAL(I-1)
      XHI=NVAL(I)
      XNN=NN
      THI=T(I-1)
      TLO=T(I)
      TO5 = TLO + ( ((XHI-XNN)/(XHI-XLO))*(THI-TLO) )
      GO TO 20
   13 TO5=T(I)
      GO TO 20
   15 CONTINUE
   20 RETURN
      END
```

Program name - NORMSEP

Purpose

Separates length frequency sampling distributions into component normal distributions. Used to estimate age group relative abundance in length samples of unageable species.

Programmer - Victor Hasselblad
Modified by Patrick K. Tomlinson

Originating organization - University of Washington and California Department of Fish and Game

Approximate storage - 20,000 (CDC 6600)

Under the assumption that the lengths of fish, within age groups, are normally distributed and that an unbiased sample of the length distribution can be obtained, this program separates the mixture of normal length distribution into its components. The method is statistically superior to graphical procedures.

If N fish were measured for length (let x = length) and it is assumed that these N fish were taken from a mixture of K normally distributed age groups (e.g. the length distribution for each age group is assumed to be normally distributed with its own mean (x_i) and standard deviation (s_i); i = 1, 2, ..., K, then the program (NORMSEP) will estimate a mean length for each age group, a standard deviation for each age group, and an expected value for each observed value. It is necessary to put the N measurements into n observed frequencies (y_j; j = 1, 2, ..., n) by partitioning the x axis (length) into n equally spaced intervals such that y_j is the number of fish measured which fall between x_{j-1} and x_j (x_0 = left end of first interval, x_1 = right end of first interval or left end of second interval, etc.

Also, the program will produce estimates of the percent of N each age group and the number of fish in N from each age group. Output includes a plotted histogram, the observed frequences, and all estimated values. The value of K may be from 1 to 10.

If each age group has an obvious mode in the histogram of y_j on x, the estimates of x_j may be assumed to be reasonable. If the preceding is true and the assumption of normality is not too bad, then the other estimates produced are not likely to be bad. However, many histograms of length distributions involving several age groups do not have obvious modes. In this latter case the problem may still be reasonably resolved if the length distribution for each age group has been described from other sources such as ageing by scales or bony parts.

For the reason mentioned in the preceding paragraph, the original program was modified to allow the user to place upper and lower bounds on the means and standard deviations. Although Hasselblad (1966) described the statistical properties of the original fitting procedure, the problems introduced by placing bounds on the estimates has not been looked at analytically. It was merely done as an empirical convenience. Anyone desiring to maintain the rigor of Hasselblad's original procedure only needs to place extremely wide bounds. Under any circumstances, Hasselblad (1966) and Cohen (1966) should be reviewed before proceeding. Also, maximum iterations were reduced to 125.

This program consists of a main program (NORMSEP), 4 function subprograms (CHISQ, NORM, TRUNC, GTRUN), 1 subroutine subprogram (VPLOT), and 9 library functions (MAXO, MINO, ALOG10, AMAX1, EXP, SQRT, ALOG, FLOAT, ABS).

The cards needed for this program are described as sets.

Set #1	1 card	FORMAT (3I5)
	Cols. 1-5	Number of intervals on x axis (number of frequencies, n). Right-justify with no decimal.
	Cols. 6-10	Number of age groups ($K \leq 10$ = number of modes expected). Right-justify with no decimal.
	Cols. 11-15	Number of sets of cut-off points (e.g. number of repetitions of set 6 through 10). Right-justify with no decimal.

Set #2	1 card	FORMAT (2F10.0)
	Cols. 1–10	Value (x_0) corresponding to the left end of the first interval. Punch decimal.
	Cols. 11–20	Length of the intervals on the x axis (e.g. length = $x_1-x_0 = x_2-x_1 = \ldots x_n-x_{n-1}$). Punch decimal.
Set #3	$1 \leq$ cards ≤ 100	FORMAT (10F5.0)
	Cols. 1–50	Values of the observed frequencies (y_j). Punch 10 values to a card in order of $j = 1,2,\ldots,n$ with each using 5 columns (including the decimal point if used).
Set #4	1 card	FORMAT (50A1)
	Cols. 1–50	This is any alphanumeric description to be printed as the vertical (y) axis of the histogram. Center on column 25.
Set #5	1 card	FORMAT (8A6)
	Cols. 1–48	This is any alphanumeric description to be printed as the horizontal (x) axis of the histogram. Left-justify.
Set #6	1 card	FORMAT (9F5.0)
	Cols. 1–45	Values of the cut-off points (K-1 values). 5 columns for each value (including decimal point if used). Punch in order $j = 1,2,\ldots,K-1$. If in doubt, choose values half way between the expected means.
Set #7	1 card	FORMAT (10F5.0)
	Cols. 1–50	Lower bounds for the means (x_j) for each age group (mode). 5 columns for each value (including decimal point if used). Punch in order $j = 1,2,\ldots,K$. If in doubt, be liberal.
Set #8	1 card	FORMAT (10F5.0)
	Cols. 1–50	Like set 7, except these are upper bounds of the means.
Set #9	1 card	FORMAT (10F5.0)
	Cols. 1–50	Like set 7, except these are the lower bounds for the standard deviations. If in doubt, divide the expected range by 8 or 10.
Set #10	1 card	FORMAT (10F5.0)
	Cols. 1–50	Like set 7, except these are upper bounds for the standard deviations. If in doubt, divide the expected range by 2 or 3.

** If you wish to try different sets of cut-off points and/or lower-upper bounds for means and/or lower-upper bounds for standard deviations, you must repeat all cards in set 6 through 10. Punch the proper number of repetitions in columns 11–15 of set 1.

*** To estimate with frequency data for more than one histogram, repeat all sets 1 through 10 for each such data set.

NO ERROR MESSAGES.

Special notes

This program reads from unit 5 and writes on unit 6. An end-of-file check occurs at statement 3+2 of the main program. Satisfactory operation should occur if the end-of-file check is removed. The logical marking statement, LL = LL.AND. 7777777777777000B, which appears at 31+2

should generally be removed for use on most computers. This was employed to reduce the accuracy of an iteration process to that of a 36-bit word size when running on a 48-bit word size machine. Excessive iteration may occur if a similar mask is not employed on computers with word sizes greater than 36 bits.

References

Cohen, A.C., Estimation of parameters for a mixture of normal distributions by V. Hasselblad. 1966 Technometrics, 8(3):445-446.

Hasselblad, V., Estimation of parameters for a mixture of normal distributions. Technometrics, 1966 8(3):431-444.

NORMSEP

```
      PROGRAM NORMSEP
C     NORMAL DISTRIBUTION SEPARATOR USING STEEPEST DESCENT METHOD
      REAL LL, MU,LL1, MUP
      REAL NORM
      DIMENSION Q(1000,10), QQ(1000), P(20), MU(20), SIG(20), F(1000),
     1 FS(1000), NF(1000), NS(20), XTIT(8), YTIT(50),PP(20),MUP(20),SIGP
     2(20), NX(20)
      DIMENSION BU(20),BL(20),AU(20),AL(20),FX(20)
      EQUIVALENCE (NF,FS)
      ALOG10(X)=TENLOGF(X)
      CONST = .39894228
      NT = 6
 2    CONTINUE
    3 FORMAT (3I5)
      READ (5,3) N,K,LCT
      IF(EOF,5) 1001,1000
1000  CONTINUE
C     N IS THE NO. OF FREQUENCIES,  K IS THE NO. OF DISTRIBUTIONS,  AND
C     LCT IS THE NO. OF SETS OF CUT-OFF POINTS INCLUDED
      LCT = MAX0(1,LCT)
      K = MIN0(K,10)
      READ (5,4) XINT, XLEN
 4    FORMAT(6F10.0)
C     XINT IS THE VALUE OF THE LEFT HAND INTERVAL
C     XLEN IS THE LENGTH OF THE INTERVALS
      READ (5,5)(F(I),I=1,N)
 5    FORMAT(10F5.0)
C     F(I) ARE THE FREQUENCIES
      FSUM =0.
      DO 6 I = 1,N
      FSUM = FSUM + F(I)
 6    NF(I) = F(I)+ .5
      READ (5,7) YTIT
 7    FORMAT(50A1)
      READ (5,8) XTIT
 8    FORMAT(8A6)
C     YTIT IS THE VERTICAL LABEL APPEARING IN VPLOT
C     XTIT IS THE HORIZONTAL ONE
      IF(N.GT.120) GO TO 13
      CALL VPLOT(NF,N,YTIT,XTIT)
      WRITE (6,9) XINT,XLEN
    9 FORMAT(27H VALUE OF LEFT-HAND LIMIT =, F10.5,5X23H, LENGTH OF INTE
     1RVALS = ,F10.5)
 13   CONTINUE
      DO 100 LYRY = 1, LCT
      READ 33,(FX(J),J=1,K)
C     NX(J) ARE THE CUT-OFF POINTS
      READ 33,(BL(J),J=1,K)
      READ 33,(BU(J),J=1,K)
      READ 33,(AL(J),J=1,K)
      READ 33,(AU(J),J=1,K)
   33 FORMAT (16F5.0)
C     BL(J) ARE LOWER BOUNDS OF MEANS
C     BU(J) ARE UPPER BOUNDS OF MEANS
C     AL(J) ARE LOWER BOUNDS OF STANDARD DEVIATIONS
C     AU(J) ARE UPPER BOUNDS OF STANDARD DEVIATIONS
      DO 34 J=1,K
      NX(J)=((FX(J)-XINT)/XLEN)+.5
      BL(J)=(BL(J)-XINT)/XLEN
      BU(J)=(BU(J)-XINT)/XLEN
      AL(J)=AL(J)/XLEN
```

```
   34 AU(J)=AU(J)/XLEN
      NX(K) = N
      DO 14 I = 1,N
   14 FS(I) = F(I)
      NB = 1
C     COMPUTE INITIAL ESTIMATES USING HALD*S PROCEDURE
      ASUM = 0.
      DO 16 J = 1,K
      NE = NX(J)
      XB = NE
      A = 0.
      B = 0.
      C = 0.
      DO 15 I = NB,NE
      CI = XB - FLOAT(I) + .5
      CF = FS(I)
      A = A + CF
      B = B + CF*CI
      C = C + CF*CI**2
   15 CONTINUE
      ZZQ=A*C/(2.*B**2)
      Z=TRUNC(ZZQ)
      G = GTRUN(Z)
      SIG(J) = B*G/A
      IF(SIG(J).GT.AU(J)) SIG(J)=AU(J)
      IF(SIG(J).LT.AL(J)) SIG(J)=AL(J)
      AVE = -Z*SIG(J)
      IF(AVE.GT.BU(J)) AVE=BU(J)
      IF(AVE.LT.BL(J)) AVE=BL(J)
      IF(ZZQ.GE..55) GO TO 39
      AVE = B/A
      IF(AVE.GT.BU(J)) AVE=BU(J)
      IF(AVE.LT.BL(J)) AVE=BL(J)
      SIG(J) = SQRT((C-B*B/A)/A)
      IF(SIG(J).GT.AU(J)) SIG(J)=AU(J)
      IF(SIG(J).LT.AL(J)) SIG(J)=AL(J)
   39 CONTINUE
      PKQ = NORM(AVE/SIG(J))
      P(J) = A/PKQ
      ASUM = ASUM + P(J)
      DO 17 KK = 1,N
      II = KK + NE
      IF(II.GT.N) GO TO 18
      ULIM = (AVE + FLOAT(KK))/SIG(J)
      BLIM = (AVE + FLOAT(KK-1))/SIG(J)
      PKK = NORM (ULIM) - NORM (BLIM)
      PKK = A*PKK/PKQ
C     SUBTRACT ESTIMATED JTH SAMPLE FROM REMAINING ONES
      FS(II) =AMAX1(FS(II)-PKK,0.)
      IF( PKK .LT.0.05) GO TO 18
   17 CONTINUE
   18 CONTINUE
      NB = NE+1
      MU(J) = FLOAT (NX(J)) - AVE
      IF(MU (J).GT.BU(J)) MU (J)=BU(J)
      IF(MU (J).LT.BL(J)) MU (J)=BL(J)
   16 CONTINUE
      DO 19 J= 1,K
   19 P(J) = P(J)/ASUM
      LL1=0.
```

```
C      COMPUTE LIKELIHOOD FUNCTION LL1
       DO 21 I = 1,N
       QQ(I) = 0.
       DO 20 J= 1,K
       Q(I,J) = CONST*EXP( -(FLOAT(I)-MU(J)-.5)**2/(2.*SIG(J)**2))/SIG(J)
 20    QQ(I)=QQ(I) + Q(I,J)* P(J)
       LL1 = LL1 + F(I)*ALOG10(QQ(I))
 21    CONTINUE
C      START METHOD OF STEEPEST DESCENT
       DO 99 IT=1,125
       LLL = IT
       PSUM = 0.
       DO 23 J= 1,K
       XDEN=0.
       SFX=0.
       SFXX=0.
       DO 24 I= 1,N
       TEMP = F(I)*Q(I,J)/QQ(I)
       XDEN = XDEN+ TEMP
       CI = FLOAT(I) -.5
       CF = CI - MU(J)
       SFX = SFX + TEMP*CI
       SFXX = SFXX + TEMP*CF**2
 24    CONTINUE
       MUP(J) = SFX/XDEN
       IF(MUP(J).LT.BL(J)) MUP(J)=BL(J)
       IF(MUP(J).GT.BU(J)) MUP(J)=BU(J)
       SIGP(J) = SQRT(SFXX/XDEN)
       IF(SIGP(J).GT.AU(J)) SIGP(J)=AU(J)
       IF(SIGP(J).LT.AL(J)) SIGP(J)=AL(J)
       PP(J) = XDEN*P(J)
       PSUM = PSUM + PP(J)
 23    CONTINUE
       DO 26 J =1, K
 26    PP(J) = PP(J)/PSUM
       LL = 0.
       DO 31 I = 1, N
       QQ(I) = 0.
       DO 32 J = 1, K
       Q(I,J)=CONST*EXP(-(FLOAT(I)-MUP(J)-.5)**2/(2.*SIGP(J)**2))/SIGP(J)
 32    QQ(I) = QQ(I) + Q(I,J)*PP(J)
       LL = LL + F(I)*ALOG10(QQ(I))
 31    CONTINUE
C      TEST FOR IMPROVED LIKELIHOOD FUNCTION LL
       LL=LL.AND.7777777777777000B
       IF(LL.LE.LL1) GO TO 30
       DO 25 J = 1, K
       P(J) = PP(J)
       MU(J) = MUP(J)
       SIG(J) = SIGP(J)
 25    CONTINUE
       LL1 = LL
 99    CONTINUE
 30    CONTINUE
       LLL = LLL - 1
       DO 37 I = 1, N
       QQ(I) = 0.
       DO 38 J = 1, K
       Q(I,J)=NORM((FLOAT(I)-MU(J))/SIG(J))-NORM((FLOAT(I)-MU(J)-1.)/
      1 SIG(J))
 38    QQ(I)=QQ(I) + Q(I,J)* P(J)
       FS(I) = QQ(I)*FSUM
```

```
 37   CONTINUE
      QQ(1) = QQ(1) + NORM(-MU(1)/SIG(1))
      QQ(N) = QQ(N) + NORM((MU(K)-FLOAT(N))/SIG(K))
      FS(1) = QQ(1)*FSUM
      FS(N) = QQ(N)*FSUM
      NN = MU(1) +1.
      PPR = 0.
C     COLLAPSE CHI-SQUARE TABLE
      DO 200 I = 1,NN
      II = I
      PPR = PPR + FS(I)
      IF(PPR-5.)200,201,201
 200  CONTINUE
 201  NLOW = II
      NN = FLOAT (N)-MU(K) + 1.
      PPS = 0.
      DO 203 I = 1, NN
      NCUR = N - I + 1
      PPS = PPS + FS(NCUR)
      IF(PPS-5.)203,204,204
 203  CONTINUE
 204  NTOP = NCUR
      AFR = 0.
      DO 210 I = 1,NLOW
 210  AFR = AFR + F(I)
      CHIS = (AFR-PPR)**2/PPR
      NHH = NTOP - 1
      NLL = NLOW + 1
      DO 211 I = NLL,NHH
 211  CHIS = CHIS + (F(I)-FS(I))**2/FS(I)
      AFR = 0.
      DO 213 I = NTOP,N
 213  AFR = AFR + F(I)
      CHIS = CHIS + (AFR-PPS)**2/PPS
      NDEG = NTOP - NLOW - 3*K+1
C     WRITE OUT RESULTS
      WRITE (NT,419) XTIT
 419  FORMAT(1H1,20X,8A6/1H0)
      WRITE (NT,413)
413   FORMAT(38H RESULTS USING STEEPEST DESCENT METHOD)
      WRITE (NT,401) LLL
 401  FORMAT(13H VALUES AFTER ,I5,11H ITERATIONS)
      WRITE (NT,416)
 416  FORMAT(6H GROUP,5X4HMEAN,8X,8HST. DEV.,5X,7HPERCENT,4X,4HSIZE)
      DO 402 J = 1, K
      SSG = (SIG(J)*XLEN)
      XAVE = MU(J)*XLEN+XINT
      PSZ = P(J)*FSUM
      PPS = P(J)*100.
      WRITE(NT,94) J,XAVE,SSG,PPS,PSZ
 94   FORMAT(1H0,I3,F13.5,F14.6,F11.2,F9.1)
 402  CONTINUE
      NSUM = FSUM + .5
      WRITE(NT,414) NSUM
 414  FORMAT(20H TOTAL SAMPLE SIZE = I7)
      WRITE(NT,405)
405   FORMAT(1H0,20X,32HACTUAL VS. PREDICTED FREQUENCIES)
      NAZ = (K-1)/10 + 1
      DO 406 I = 1, NAZ
      NQ1 = I*10 - 9
      NQ2 = MIN0 (I*10,N)
      WRITE(NT,407) (F(J),J=NQ1,NQ2)
      WRITE(NT,408) (FS(J),J=NQ1,NQ2)
```

```
 407  FORMAT(1H0,10F10.1)
 408  FORMAT(1H ,10F10.1)
 406  CONTINUE
      FX(K)=NX(K)*XLEN+XINT
      WRITE (NT,404) (FX(J),J=1,K)
 404  FORMAT (1H0/1H0,14HCUT-OFF POINTS,21X,10(F5.0,2X))
   81 FORMAT (1H ,25HLOWER BOUNDS FOR MEANS    ,10X,10(F5.0,2X))
   82 FORMAT (1H ,25HUPPER BOUNDS FOR MEANS    ,10X,10(F5.0,2X))
   83 FORMAT (1H ,25HLOWER BOUNDS FOR ST. DEV.,10X,10(F5.0,2X))
   84 FORMAT (1H ,25HUPPER BOUNDS FOR ST. DEV.,10X,10(F5.0,2X))
      DO 85 J=1,K
      BL(J)=BL(J)*XLEN+XINT
      BU(J)=BU(J)*XLEN+XINT
      AL(J)=AL(J)*XLEN
   85 AU(J)=AU(J)*XLEN
      WRITE (NT,81) (BL(J),J=1,K)
      WRITE (NT,82) (BU(J),J=1,K)
      WRITE (NT,83) (AL(J),J=1,K)
      WRITE (NT,84) (AU(J),J=1,K)
      WRITE(NT,403) NLOW,NTOP
403   FORMAT(25H0LOWER COLLAPSING POINT =,I3,10X,24HUPPER COLLAPSING POI
     1NT = ,I3)
      WRITE(NT,410) NDEG
 410  FORMAT(21H DEGREES OF FREEDOM = I3)
      SIGN = CHISQ(CHIS,NDEG)
      WRITE(NT,409) CHIS, SIGN
 409  FORMAT(19H CHI SQUARE VALUE = F8.3,10X,7HPROB. = F6.3)
      WRITE(NT,411) LL1
 411  FORMAT(20H LOG OF LIKELIHOOD = E16.8)
 100  CONTINUE
      GO TO 2
1001  STOP
       END


      FUNCTION CHISQ(X,N)
C     EXACT FORMULA - SHARE DISTRIBUTION NO. 528.
      REAL NORM
      SUM = 0.
      IF(X.GT.0.0.AND.N.GT.0)GO TO 2
      CHISQ = 1.
      RETURN
2     CONTINUE
      IF((N/2)*2.NE.N)GO TO 11
      IF(N.LE.2)GO TO 3
      M = N/2 - 1
      TERM = 1.
      DO 4 I = 1, M
      CI = 2.*FLOAT(I)
      TERM = TERM*X/CI
      SUM = SUM + TERM
 4    CONTINUE
 3    CHISQ = EXP(-X/2.0)*(1.0+SUM)
      RETURN
 11   CONTINUE
      TERM = 1.0/X
      IF(N.LE.1)GO TO 12
      M = (N-1)/2
      DO 13 I = 1, M
      CI = 2.*FLOAT(I)-1.
      TERM = TERM*X/CI
      SUM = SUM + TERM
 13   CONTINUE
 12   XS = SQRT(X)
      CHISQ = 2.0-2.*NORM( XS)+ XS*EXP(-X/2.)*.7978846*SUM
      RETURN
      END
```

NORMSEP

```
      FUNCTION GTRUN(X)
C     G(X) - HALD - MAX-LIKE EST. - ANNALS MATH STAT. VOL. 24,P. 557
C     USES APPROX. USED IN NORM.
      DATA(S2P=1.12837917),(P=.3275911),(A1=.225836846),(A2=-.252128668)
     1,(A3=1.25969513),(A4=-1.28782245),(A5=.94064607),(S2=.707106781)
      XN = 1./(1. + P*ABS(X*S2))
      PHI = S2P*EXP(-X**2/2.)
      PHIS=1.-XN*(A1+XN*(A2+XN*(A3+XN*(A4+XN*A5))))*PHI
      TERM = .5*PHIS
      IF(X.GT.0.) TERM = -TERM
      GTRUN = 1./(PHI/(2.82842712*(TERM+.5))-X)
      RETURN
      END



      FUNCTION NORM(X)
      REAL NORM
C     APPROX. FOR DIG. COMP. - HASTINGS, P. 169
      DATA(S2P=1.12837917),(P=.3275911),(A1=.225836846),(A2=-.252128668)
     1,(A3=1.25969513),(A4=-1.28782245),(A5=.94064607),(S2=.707106781)
      XN = 1./(1. + P*ABS(X*S2))
      PHI = S2P*EXP(-X**2/2.)
      PHIS=1.-XN*(A1+XN*(A2+XN*(A3+XN*(A4+XN*A5))))*PHI
      TERM = .5*PHIS
      IF(X.LT.0.) TERM = -TERM
      NORM = .5 + TERM
      RETURN
      END



      FUNCTION TRUNC(X)
      DATA(A0=34.758138),(A1=-69.16544),(A2=38.338991),(A3=-5.381936),(B
     11=5.01975036),(B2=-7.0780578),(B3=1.6336409)
      Z = SQRT(-2.*ALOG(X-.5))
      TRUNC = (((A3*Z+A2)*Z+A1)*Z+A0)/(((B3*Z+B2)*Z+B1)*Z+1.)
      RETURN
      END



      SUBROUTINE VPLOT(NF,N,YTIT,XTIT)
CVPLOT         FREQUENCY PLOTTER
      DIMENSION XALPH(120),XALP(120),NF(1),YTIT(1),XTIT(1)
      DATA(XPL=6H......),(BL=1H ),(XLET=1HX)
      IF(N-120)54,54,55
 54   CONTINUE
      NMAX = 0
      DO 11 I = 1, N
      NMAX = MAX0 (NMAX,NF(I))
 11   CONTINUE
      NL = (N+4)/5
      NAQ = NL*5
      NQQ = MIN0 (NAQ,50)
      NSC = (NMAX+NQQ-1)/NQQ
      ISC = NSC
      ND = NSC/2
      DO 56 I = 1, NAQ
      XALP(I) = XPL
```

```
 56   XALPH(I) = BL
      DO 74 I = 5, NAQ, 5
      XALPH(I) = XPL
 74   CONTINUE
      WRITE              (6, 62)
62    FORMAT(1H1,20X,12HPLOT OF DATA,/1H )
      DO 50 I = 1, N
 50   NF(I) = (NF(I) +ND)/NSC
      DO 51 I = 5, 50, 5
      KK = 55 - I
      DO 57 JA = 1, N
      IF(NF(JA)-KK)57,59,57
 59   XALPH(JA) = XLET
      XALP(JA) = XLET
 57   CONTINUE
      KI = KK*ISC
      WRITE              (6,53)YTIT(I-4),KI,XPL,(XALP(JJ),JJ=1,NAQ)
   53 FORMAT (1H ,4X,A1,I5,109A1)
      DO 51 J = 2, 5
      KB = 56-I-J
      DO 60 JA = 1, N
      IF(NF(JA)-KB)60,61,60
 61   XALPH(JA) = XLET
      XALP(JA) = XLET
 60   CONTINUE
      KC = I + J - 5
      WRITE              (6,52)YTIT(KC),XPL,(XALPH(JJ),JJ=1,NAQ)
   52 FORMAT (1H ,4X,A1,5X,109A1)
 51   CONTINUE
      WRITE              (6, 64) XPL, (XPL,I=1,NL)
   64 FORMAT (1H ,10X,A1,21A5)
      WRITE              (6, 65) (I,I=5,NAQ,5)
   65 FORMAT (1H ,11X,21I5)
      WRITE(6,73)(XTIT(I),I=1,8),NSC
 73   FORMAT(1H0,8A6,5X, 18HY VALUES SCALED BY I3)
 55   CONTINUE
      RETURN
      END
```

RESOURCE MATERIALS COLLECTION
East-West Food Institute
East-West Center
1777 East-West Road
Honolulu, Hawaii 96822

Library
NOAA National Marine Fisheries Service
SWFSC Honolulu Laboratory F/SWC2
2570 Dole Street
Honolulu, HI 96822-2396

WI/B2939/3.71/E/1/1000